```
RRRRRRRRRRRR        UUU          UUU  NNN        NNN    000000000    FFFFFFFFFFFFFFF    FFFFFFFFFFFFFFF
RRRRRRRRRRRR        UUU          UUU  NNN        NNN    000000000    FFFFFFFFFFFFFFF    FFFFFFFFFFFFFFF
RRRRRRRRRRRR        UUU          UUU  NNN        NNN    000000000    FFFFFFFFFFFFFFF    FFFFFFFFFFFFFFF
RRR      RRR        UUU          UUU  NNN        NNN   000      000   FFF               FFF
RRR      RRR        UUU          UUU  NNN        NNN   000      000   FFF               FFF
RRR      RRR        UUU          UUU  NNN        NNN   000      000   FFF               FFF
RRR      RRR        UUU          UUU  NNNNNN     NNN   000      000   FFF               FFF
RRR      RRR        UUU          UUU  NNNNNN     NNN   000      000   FFF               FFF
RRRRRRRRRRRR        UUU          UUU  NNN  NNN   NNN   000      000   FFFFFFFFFFF        FFFFFFFFFFF
RRRRRRRRRRRR        UUU          UUU  NNN  NNN   NNN   000      000   FFFFFFFFFFF        FFFFFFFFFFF
RRRRRRRRRRRR        UUU          UUU  NNN  NNN   NNN   000      000   FFFFFFFFFFF        FFFFFFFFFFF
RRR   RRR           UUU          UUU  NNN    NNNNN     000      000   FFF               FFF
RRR    RRR          UUU          UUU  NNN    NNNNN     000      000   FFF               FFF
RRR     RRR         UUU          UUU  NNN      NNN     000      000   FFF               FFF
RRR      RRR        UUU          UUU  NNN      NNN     000      000   FFF               FFF
RRR       RRR       UUU          UUU  NNN      NNN     000      000   FFF               FFF
RRR        RRR      UUUUUUUUUUUUUUUU  NNN      NNN      000000000     FFF               FFF
RRR         RRR     UUUUUUUUUUUUUUUU  NNN      NNN      000000000     FFF               FFF
RRR         RRR     UUUUUUUUUUUUUUUU  NNN      NNN      000000000     FFF               FFF
```

FILEID**CLH

```
CCCCCCCC  LL           HH        HH
CCCCCCCC  LL           HH        HH
CC        LL           HH        HH
CC        LL           HH        HH
CC        LL           HH        HH
CC        LL           HH        HH
CC        LL           HHHHHHHHHH
CC        LL           HHHHHHHHHH
CC        LL           HH        HH
CC        LL           HH        HH
CC        LL           HH        HH    ....
CC        LL           HH        HH    ....
CCCCCCCC  LLLLLLLLLL   HH        HH    ....
CCCCCCCC  LLLLLLLLLL   HH        HH    ....
```

```
LL            IIIIII       SSSSSSSS
LL            IIIIII       SSSSSSSS
LL              II       SS
LL              II       SS
LL              II       SS
LL              II         SSSSSS
LL              II         SSSSSS
LL              II               SS
LL              II               SS
LL              II               SS
LL              II               SS
LLLLLLLLLL    IIIIII       SSSSSSSS
LLLLLLLLLL    IIIIII       SSSSSSSS
```

```
0001   0  %TITLE 'file processing interface and command line handler'
0002   0  MODULE clh (       IDENT = 'V04-000'
0003   0                     %BLISS32 [, ADDRESSING_MODE (EXTERNAL    = long_relative,
0004   0                                                  NONEXTERNAL = long_relative)]
0005   0                     ) =
0006   1  BEGIN
0007   1
0008   1  !********************************************************************************
0009   1  !*                                                                              *
0010   1  !*    COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                   *
0011   1  !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                      *
0012   1  !*  ALL RIGHTS RESERVED.                                                        *
0013   1  !*                                                                              *
0014   1  !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED       *
0015   1  !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE       *
0016   1  !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER       *
0017   1  !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY       *
0018   1  !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY      *
0019   1  !*  TRANSFERRED.                                                                *
0020   1  !*                                                                              *
0021   1  !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE       *
0022   1  !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT       *
0023   1  !*  CORPORATION.                                                                *
0024   1  !*                                                                              *
0025   1  !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS       *
0026   1  !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                     *
0027   1  !*                                                                              *
0028   1  !*                                                                              *
0029   1  !********************************************************************************
0030   1
0031   1  !++
0032   1  ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
0033   1  !
0034   1  ! ABSTRACT:      File processing interface and command line handler.
0035   1  !
0036   1  ! ENVIRONMENT: Transportable
0037   1  !
0038   1  ! AUTHOR:        R.W.Friday      CREATION DATE:  April, 1978
0039   1  !
```

```
    41      0040  1 %SBTTL 'Revision History'
    42      0041  1 !
    43      0042  1 ! MODIFIED BY:
    44      0043  1 !
    45      0044  1 !      041     REM00041        Ray Marshall    23-February-1984
    46      0045  1 !              Moved IN_TYPE and OUT_TYPE to GLBDAT.BLI and renamed them to
    47      0046  1 !              IPFTOP and OPFTOP respectivly.  Also created there another
    48      0047  1 !              global variable and one global literal to support new logic
    49      0048  1 !              herein.  All this was done because in VMS V4 there is a new
    50      0049  1 !              feature (search lists) which necessatates our making calls
    51      0050  1 !              to LIB$FIND_FILE to find input files after the first one
    52      0051  1 !              when multiple input files are specified.  To do this, the
    53      0052  1 !              logic to verify the input filespec was copied to a new
    54      0053  1 !              routine within RUNOFF.BLI and changes were made herein to
    55      0054  1 !              the sections that open the output and input files.
    56      0055  1 !
    57      0056  1 !      040     KFA00040        Ken Alden       20-Jul-1983
    58      0057  1 !              Fixed wild card bug interaction with /auto.
    59      0058  1 !
    60      0059  1 !      039     KFA00039        Ken Alden       10-Jun-1983
    61      0060  1 !              Improved error handling of wild-carding input file names.
    62      0061  1 !
    63      0062  1 !      038     KFA00038        Ken Alden       4-May-1983
    64      0063  1 !              Added conditional for gca_skip_out in OUT_NO_CRLF branch.
    65      0064  1 !
    66      0065  1 !      037     KAD00037        Keith Dawson    14-Apr-1983
    67      0066  1 !              Fixed bug: FLIP (.BFL) output file was not being opened.
    68      0067  1 !
    69      0068  1 !      036     KAD00036        Keith Dawson    22-Mar-1983
    70      0069  1 !              Added LN01 support.
    71      0070  1 !
    72      0071  1 !      035     KFA00035        Ken Alden       07-Mar-1983
    73      0072  1 !              Global edit of all modules. Updated module names, idents,
    74      0073  1 !              copyright dates. Changed require files to BLISS library.
    75      0074  1 !
    76      0075  1 !--
```

CLH
V04-000

K 14
file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page   3
Module Level Declarations   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (3)

```
 78    0076  1  %SBTTL 'Module Level Declarations'
 79    0077  1  !
 80    0078  1  ! TABLE OF CONTENTS:
 81    0079  1  !
 82    0080  1  FORWARD ROUTINE
 83    0081  1      GET_OUT_DEFAULT : NOVALUE,
 84    0082  1      BWAIT : NOVALUE,
 85    0083  1      FBWAIT : NOVALUE;
 86    0084  1
 87    0085  1  !
 88    0086  1  ! INCLUDE FILES:
 89    0087  1  !
 90    0088  1
 91    0089  1  LIBRARY 'NXPORT:XPORT';                    ! XPORT Library
 92    0090  1  REQUIRE 'REQ:RNODEF';                      ! RUNOFF variant definitions
 93    0221  1
 94  U 0222  1  %IF DSRPLUS %THEN
 95  U 0223  1  LIBRARY 'REQ:DPLLIB';                      ! DSRPLUS BLISS Library
 96    0224  1  %ELSE
 97    0225  1  LIBRARY 'REQ:DSRLIB';                      ! DSR BLISS Library
 98    0226  1  %FI
 99    0227  1
100    0228  1  !
101    0229  1  ! MACROS:
102    0230  1  !
103    0231  1  MACRO
104  M 0232  1      XPROMPT (TEXT) =
105  M M 0233 1          ($XPO_PUT( IOB = TTOIOB
106  M M 0234 1          ,STRING = ( %CHARCOUNT(TEXT)
107    0235  1          ,CH$PTR(UPLIT(TEXT))) )   )%;
108    0236  1
109    0237  1  !
110    0238  1  ! OWN STORAGE:
111    0239  1  !
112    0240  1  OWN
113    0241  1      status;
114    0242  1
115    0243  1  OWN
116    0244  1      DEF_OUT_LNG,
117    0245  1      DEF_OUT_SPC : VECTOR [CH$ALLOCATION (50)];
118    0246  1
119    0247  1  !
120    0248  1  ! EXTERNAL REFERENCES:
121    0249  1  !
122    0250  1  EXTERNAL
123    0251  1      fra : FIXED_STRING,
124    0252  1      gca : GCA_DEFINITION,
125    0253  1      fs01 : FIXED_STRING,
126    0254  1      ira : FIXED_STRING,
127    0255  1      irac : IRAC_DEFINITION,
128    0256  1      ipftyp,                                ! Cell in GLBDAT to hold the index into IPFTOP
129    0257  1      ipftop : VECTOR,
130    0258  1      opftop : VECTOR;
131    0259  1
132    0260  1  EXTERNAL
133    0261  1      IOBSTK : BLOCK,                        !IOB stack for doing .REQUIRE.
134    0262  1      RNEIOB : REF $XPO_IOB (),              !Always points to IOB for primary input.
```

```
: 135    0263 1    RNIIOB : REF $XPO_IOB (),                    !Primary input file
: 136    0264 1    RNOIOB : REF $XPO_IOB (),                    !IOB for output file
: 137    0265 1    TSIIOB : $XPO_IOB (),                        !IOB for STREAM input from terminal.
: 138    0266 1    TTIIOB : $XPO_IOB (),                        !IOB for input from terminal
: 139    0267 1    TTOIOB : $XPO_IOB ();                        !IOB for output to terminal
: 140    0268 1
: 141    0269 1  EXTERNAL LITERAL
: 142    0270 1    ipftct,      ! Literal defining the lengths of IPFTOP and OPFTOP.
: 143    0271 1
: 144    0272 1              ! Error messages
: 145    0273 1    rnfile,
: 146  U 0274 1  %IF dsrplus %THEN
: 147  U 0275 1    rnfoft,
: 148    0276 1  %FI
: 149    0277 1    rnfrtl;
: 150    0278 1
: 151    0279 1  EXTERNAL ROUTINE
: 152    0280 1    ERMS,
: 153  U 0281 1  %IF DSRPLUS %THEN
: 154  U 0282 1    ERM,
: 155    0283 1  %FI
: 156    0284 1    GRAB_RESULTANT,
: 157    0285 1    PUTMSG,
: 158    0286 1    TSTTFE;
: 159    0287 1
```

M 14

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 5
V04-000   CLH -- OPCODE controlled main-line routine   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (4)

```
 161    0288  1  %sbttl 'CLH -- OPCODE controlled main-line routine'
 162    0289  1  GLOBAL ROUTINE clh (opcode) =
 163    0290  1
 164    0291  1  !++
 165    0292  1  ! FUNCTIONAL DESCRIPTION:
 166    0293  1  !
 167    0294  1  !     See ABSTRACT, above.
 168    0295  1  !
 169    0296  1  ! FORMAL PARAMETERS:
 170    0297  1  !
 171    0298  1  !     OPCODE specifies the operation to be performed.
 172    0299  1  !
 173    0300  1  ! IMPLICIT INPUTS:      None
 174    0301  1  !
 175    0302  1  ! IMPLICIT OUTPUTS:     None
 176    0303  1  !
 177    0304  1  ! ROUTINE VALUE:
 178    0305  1  ! COMPLETION CODES:
 179    0306  1  !
 180    0307  1  !     See CLHCC.REQ for a description.
 181    0308  1  !
 182    0309  1  ! SIDE EFFECTS:         None
 183    0310  1  !
 184    0311  1  !--
 185    0312  1
 186    0313  2      BEGIN
 187    0314  2
 188    0315  2      LOCAL
 189    0316  2          temp;
 190    0317  2
 191    0318  2      CASE .opcode FROM 1 TO clh_ops_count OF
 192    0319  2          SET
 193    0320  2
```

N 14

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 6
V04-000   CLH -- open input file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (5)

```
:   195       0321  2  %sbttl 'CLH -- open input file'
:   196       0322  2
:   197       0323  2          [CLH_OPEN_INPUT] :                ! Open primary input file.
:   198       0324  3              BEGIN
:   199       0325  3
:   200  U    0326  3  %IF NOT %BLISS(BLISS32) AND DSRPLUS %THEN
:   201  U    0327  3
:   202  U    0328  3              INCR I FROM 0 TO (ipftct-1) DO        ! fourteen different file types
:   203  U    0329  3                  BEGIN                            ! loop until one found.
:   204  U    0330  3                  IF .gca_pass_count GTR 1
:   205  U    0331  3                  THEN
:   206  U    0332  3                      BEGIN
:   207  U    0333  3                      status = $XPO_OPEN  (IOB      = .rniiob
:   208  U    0334  3                                          ,OPTIONS = INPUT
:   209  U    0335  3                                          ,DEFAULT = (4, .ipftop [ .ipftyp ])
:   210  U    0336  3                                          ,FAILURE = grab_resultant);
:   211  U    0337  3                      IF .status
:   212  U    0338  3                      THEN
:   213  U    0339  3                          I = .ipftyp;
:   214  U    0340  3                      END
:   215  U    0341  3                  ELSE
:   216  U    0342  3                      status = $XPO_OPEN  (IOB      = .rniiob
:   217  U    0343  3                                          ,OPTIONS = INPUT
:   218  U    0344  3                                          ,DEFAULT = (4, .ipftop [.I ])
:   219  U    0345  3                                          ,FAILURE = grab_resultant);
:   220  U    0346  3
:   221  U    0347  3                  IF .status THEN EXITLOOP  ! If we open a file, exit the INCR loop
:   222  U    0348  3
:   223  U    0349  3                  END;                !end of INCR loop.
:   224       0350  3  %ELSE
:   225  P    0351  3              status = $XPO_OPEN  (IOB      = .rniiob
:   226  P    0352  3                                  ,OPTIONS = INPUT
:   227  P    0353  3                                  ,DEFAULT = (4, CH$PTR (UPLIT ('.RNO')))
:   228       0354  3                                  ,FAILURE = grab_resultant);
:   229       0355  3  %FI
:   230       0356  3              IF  .status                  ! Succeeded in opening (any type)?
:   231       0357  3              THEN
:   232       0358  4                  BEGIN                    !     Yes...
:   233       0359  4                  BIND
:   234       0360  4                      file_spec_stuff = rniiob [iob$t_resultant] : $STR_DESCRIPTOR ();
:   235       0361  4
:   236       0362  4                  ! Pick off the name and length of the file spec for ERROR.BLI:
:   237       0363  4
:   238       0364  4                  irac_fspecp = .file_spec_stuff [str$a_pointer];
:   239       0365  4                  irac_fspecc = .file_spec_stuff [str$h_length];
:   240       0366  4
:   241  U    0367  4  %IF NOT %BLISS(BLISS32) AND DSRPLUS %THEN
:   242  U    0368  4                  ipftyp = .i;
:   243  U    0369  4                  IF  (.I GTR 0)               !Succeeded in opening, but not '.RNO'.
:   244  U    0370  4                  THEN
:   245  U    0371  4                      ERM (rnfoft);
:   246       0372  4  %FI
:   247       0373  4                  RETURN clh_normal
:   248       0374  4                  END
:   249       0375  3              ELSE                ! Open failed.
:   250       0376  3                  RETURN clh_cant_open
:   251       0377  2              END;
```

```
CLH                  file processing interface and command line hand 15-Sep-1984 23:56:16    VAX-11 Bliss-32 V4.0-742        Page   7
V04-000              CLH -- open output file                          14-Sep-1984 13:05:41    DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1     (6)
```

```
 253     0378  2  %sbttl 'CLH -- open output file'
 254     0379  2
 255     0380  2          [CLH_OPEN_OUT] :
 256     0381  2          ! Open output file.
 257     0382  3          BEGIN
 258     0383  3          get_out_default (rniiob [iob$t_resultant]);
 259     0384  3
 260     0385  3          CASE .gca_op_dev FROM op_dev_first TO op_dev_last OF
 261     0386  3
 262     0387  3              SET
 263     0388  3
 264     0389  3              [op_dev_line_printer, op_dev_diablo] :
 265     0390  3
 266     0391  3                  !Normal cases: lineprinter output. Open STREAM output file using
 267     0392  3                  !file type already determined (.MEM, .MEC, or whatever).
 268     0393  3                  !
 269   P 0394  3                  status = $XPO_OPEN
 270   P 0395  3                          (IOB         = .rnoiob
 271   P 0396  3                          ,OPTIONS     = OUTPUT
 272   P 0397  3                          ,ATTRIBUTES  = STREAM
 273   P 0398  3                          ,RELATED     = rniiob [iob$t_resultant]
 274   P 0399  3                          ,DEFAULT     = (.def_out_lng, CH$PTR (def_out_spc))
 275   P 0400  3                          ,FAILURE     = grab_resultant
 276     0401  3                          );
 277   U 0402  3  %IF DSRPLUS %THEN
 278   U 0403  3              [op_dev_vt100] :
 279   U 0404  3
 280   U 0405  3                  IF NOT .gca_s_output
 281   U 0406  3                  THEN
 282   U 0407  3                          !
 283   U 0408  3                          !User said /DEC=VT100, and did not say /OUTPUT=name,
 284   U 0409  3                          !so send output to TT: (SYS$OUTPUT).
 285   U 0410  3                          !
 286   U 0411  3                          status = $XPO_OPEN
 287   U 0412  3                                  (IOB         = .rnoiob
 288   U 0413  3                                  ,OPTIONS     = OUTPUT
 289   U 0414  3                                  ,FILE_SPEC   = $XPO_OUTPUT
 290   U 0415  3                                  ,RELATED     = rniiob [iob$t_resultant]
 291   U 0416  3                                  ,DEFAULT     = (.def_out_lng, CH$PTR (def_out_spc))
 292   U 0417  3                                  ,FAILURE     = grab_resultant
 293   U 0418  3                                  )
 294   U 0419  3                  ELSE
 295   U 0420  3                          !
 296   U 0421  3                          !Open non-STREAM output file using file type already
 297   U 0422  3                          !determined (.VT1).
 298   U 0423  3                          !
 299   U 0424  3                          status = $XPO_OPEN
 300   U 0425  3                                  (IOB         = .rnoiob
 301   U 0426  3                                  ,OPTIONS     = OUTPUT
 302   U 0427  3                                  ,RELATED     = rniiob [iob$t_resultant]
 303   U 0428  3                                  ,DEFAULT     = (.def_out_lng, CH$PTR (def_out_spc))
 304   U 0429  3                                  ,FAILURE     = grab_resultant
 305   U 0430  3                                  );
 306     0431  3  %FI
 307     0432  3  %IF LN01 %THEN
 308     0433  3              [op_dev_ln01, op_dev_ln01e] :
 309     0434  3
```

```
 310        0435    3                              !Open non-STREAM output file using file type already
 311        0436    3                              !determined (.LNI).
 312        0437    3                              !
 313     P  0438    3                              status = $XPO_OPEN
 314     P  0439    3                                        (IOB          = .rnoiob
 315     P  0440    3                                        ,OPTIONS      = OUTPUT
 316     P  0441    3                                        ,RELATED      = rniiob [iob$t_resultant]
 317     P  0442    3                                        ,DEFAULT      = (.def_out_lng, CH$PTR (def_out_spc))
 318     P  0443    3                                        ,FAILURE      = grab_resultant
 319        0444    3                                        );
 320        0445    3   %FI
 321     U  0446    3   %IF FLIP %THEN
 322     U  0447    3                      [op_dev_flip] :
 323     U  0448    3                              !
 324     U  0449    3                              !Open non-STREAM output file using file type already
 325     U  0450    3                              !determined (.BFL).
 326     U  0451    3                              !
 327     U  0452    3                              status = $XPO_OPEN
 328     U  0453    3                                        (IOB          = .rnoiob
 329     U  0454    3                                        ,OPTIONS      = OUTPUT
 330     U  0455    3                                        ,RELATED      = rniiob [iob$t_resultant]
 331     U  0456    3                                        ,DEFAULT      = (.def_out_lng, CH$PTR (def_out_spc))
 332     U  0457    3                                        ,FAILURE      = grab_resultant
 333     U  0458    3                                        );
 334        0459    3   %FI
 335        0460    3                      [INRANGE] : 0;
 336        0461    3                      [OUTRANGE] : 0;
 337        0462    3
 338        0463    3                      TES;
 339        0464    3
 340        0465    3                  IF  NOT .status
 341        0466    3                  THEN
 342        0467    3                      RETURN clh_cant_open
 343        0468    3                  ELSE
 344        0469    3                      RETURN clh_normal;
 345        0470    3
 346        0471    2                  END;
```

CLH
V04-000

D 15
file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 9
CLH -- Read one record from current input file 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (7)

```
 348    0472   2 %sbttl 'CLH -- Read one record from current input file'
 349    0473   2
 350    0474   2      [CLH_READ_INPUT] :
 351    0475   2          ! Read one record from current input file.
 352    0476   3          BEGIN
 353    0477   3          status = $XPO_GET (IOB = .rniiob);
 354    0478   3
 355    0479   3          IF  .status  OR                       ! If no error in above get or
 356    0480   4              (.status EQL xpo$_truncated)      !Truncated records are really not too bad.
 357    0481   3          THEN
 358    0482   3              ! A record was successfully read. Set up information
 359    0483   3              ! needed by the remainder of the program.
 360    0484   4              BEGIN
 361    0485   4              irac_iseqn  = .rniiob [iob$g_seq_numb]; !Input record/sequence number.
 362    0486   4              irac_ipagen = .rniiob [iob$h_page_numb];        !Input page number.
 363    0487   4              irac_seqn_flag = .rniiob [iob$v_sequenced];     !Indicates meaning of IRAC_ISEQN.
 364    0488   4
 365    0489   4              ! Inform user if it was a truncated record that was read.
 366    0490   4              IF  .status EQL xpo$_truncated
 367    0491   4              THEN
 368    0492   4                  erms (rnfrtl
 369    0493   4                      ,.rniiob [iob$a_string]
 370    0494   4                      ,min (.rniiob [iob$h_string]
 371    0495   4                      ,50));
 372    0496   4
 373    0497   4              ! Set up the input as a FIXED_STRING.
 374    0498   4              fs_start (ira)   = .rniiob [iob$a_string];
 375    0499   4              fs_next (ira)    = .fs_start (ira);
 376    0500   4              fs_maxsize (ira) = .rniiob [iob$h_string];
 377    0501   4              fs_length (ira)  = .fs_maxsize (ira);
 378    0502   4
 379    0503   4 !******PATCH TO GET AROUND XPORT DEFICIENCIES
 380    0504   4 !Upon entering this block, the fixed string IRA is set up such that
 381    0505   4 !FS_NEXT(IRA) returns a CH$PTR to the first character to be processed.
 382    0506   4 !The contents of KHAR are undefined.   The block is exited with the
 383    0507   4 !same conditions holding; the only effects are:
 384    0508   4 !    1. Updating the input page/line counters, and
 385    0509   4 !    2. Movement of FS_NEXT(IRA) over all LEADING formfeeds, nulls,
 386    0510   4 !       and dels.
 387    0511   4
 388    0512   5              BEGIN
 389    0513   5
 390    0514   5              LITERAL
 391    0515   5                  ff = %O'014';
 392    0516   5
 393    0517   5              LOCAL
 394    0518   5                  ptr,
 395    0519   5                  x;
 396    0520   5
 397    0521   5              WHILE (.fs_length (ira) GTR 0) DO
 398    0522   5
 399    0523   6                  BEGIN
 400    0524   6                  ! First point to the character about to be considered.
 401    0525   6                  ptr = .fs_next (ira);
 402    0526   6                  ! Now, actually pick up the character.  Note that
 403    0527   6                  ! FS_RCHAR is not used because it advances its pointer
 404    0528   6                  ! such that if this character is not to be discarded
```

E 15

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 10
V04-000   CLH -- Read one record from current input file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (7)

```
405    0529  6                              ! we can't back up.
406    0530  6                              x = CH$RCHAR (.ptr);
407    0531  6
408    0532  6                              SELECT .x OF
409    0533  6                                  SET
410    0534  6
411    0535  6                                  [FF] :
412    0536  7                                      BEGIN
413    0537  7                                      ! If the file is sequenced, LEADING formfeeds do not
414    0538  7                                      ! start new pages; the assumption here is that the
415    0539  7                                      ! file will be looked at using the same editor (presumably SOS)
416    0540  7                                      ! as the one that created it, and that editor behaves like
417    0541  7                                      ! SOS.  The action for sequenced files is
418    0542  7                                      ! simply to ignore the formfeed.
419    0543  7                                      ! For unsequenced files, leading formfeeds do start new
420    0544  7                                      ! pages, especially if you look at that file using SOS.
421    0545  7                                      ! In such cases, XPORT does not pay attention to the
422    0546  7                                      ! formfeeds, and feeds them through without counting a
423    0547  7                                      ! new page.  In this case, WE have to look for them and
424    0548  7                                      ! set up the page and sequence number items.
425    0549  7                                      ! (Note however that XPORT does count pagemarks).
426    0550  7
427    0551  7                                      IF  NOT .rniiob [iob$v_sequenced]
428    0552  7                                      THEN
429    0553  8                                          BEGIN
430    0554  8                                          irac_ipagen = .irac_ipagen + 1;
431    0555  8                                          irac_iseqn = 1;
432    0556  8                                          rniiob [iob$g_seq_numb] = .irac_iseqn;
433    0557  8                                          rniiob [iob$h_page_numb] = .irac_ipagen;
434    0558  7                                          END;
435    0559  6
436    0560  6                                      END;
437    0561  6
438    0562  6                                  [0, FF, DEL] :
439    0563  7                                      BEGIN
440    0564  7                                      ! Actually read the character that is being rejected.
441    0565  7                                      ! This results in FS_NEXT(IRA) pointing to the next
442    0566  7                                      ! character that is to be considered.
443    0567  7                                      fs_rchar (ira, x);  ! (X is a dummy for this one line.)
444    0568  6                                      END;
445    0569  6
446    0570  6                                  [OTHERWISE] :
447    0571  6                                      EXITLOOP;
448    0572  6
449    0573  6                                  TES;
450    0574  6
451    0575  5                              END;
452    0576  4
453    0577  4                          END;
454    0578  4   !******END OF PATCH
455    0579  4
456    0580  4                          RETURN clh_normal;
457    0581  4                          END
458    0582  3                      ELSE
459    0583  4                          BEGIN
460    0584  4
461    0585  4                          IF  .status EQL xpo$_end_file
```

```
CLH                          F 15
V04-000   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 11
          CLH -- Read one record from current input file  14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1  (7)
```

```
:  462    0586  4                          THEN                                      ! End of file processing
:  463    0587  5                            BEGIN
:  464    0588  5
:  465    0589  5                            IF  .gca_req_depth NEQ 0
:  466    0590  5                            THEN
:  467    0591  5                                ! It's a ".REQUIRE" file to be closed.
:  468    0592  6                                BEGIN
:  469    0593  6                                clh (clh_pop);
:  470    0594  6                                ! Note that the following is a recursive call on this
:  471    0595  6                                ! particular code sequence.  When CLH encounters an end
:  472    0596  6                                ! of file when it attempts to read a record from a
:  473    0597  6                                ! require file, it must still, nevertheless, return a
:  474    0598  6                                ! record, unless there is not more input.  If a
:  475    0599  6                                ! .REQUIRE command is the last record read from the
:  476    0600  6                                ! file that referenced the file just closed, then the
:  477    0601  6                                ! attempt to read a record from that file will also
:  478    0602  6                                ! meet with an end of file being detected.  In this
:  479    0603  6                                ! case, you have to pop that file too, and try again.
:  480    0604  6                                ! That happens until either a record is finally read,
:  481    0605  6                                ! or all files have been popped and an end of file
:  482    0606  6                                ! occurs when trying to read the main input file.
:  483    0607  6                                RETURN clh (clh_read_input);
:  484    0608  5                                END;
:  485    0609  5
:  486    0610  5                            ! End of file on primary input file.
:  487    0611  5                            fs_length (ira) = 0;
:  488    0612  5                            RETURN clh_end_file;
:  489    0613  5                            END
:  490    0614  4                          ELSE
:  491    0615  4                          ! Error reading input file.
:  492    0616  5                            BEGIN
:  493    0617  5                            fs_length (ira) = 0;
:  494    0618  5                            RETURN clh_cant_read;
:  495    0619  4                            END;
:  496    0620  4
:  497    0621  3                          END;
:  498    0622  3
:  499    0623  2                      END;
:  500    0624  2
```

G 15

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 12
V04-000   CLH -- Write 1 record to O/P file with CRLF suf 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (8)

```
  502   0625  2  %sbttl 'CLH -- Write 1 record to O/P file with CRLF suffix'
  503   0626  2
  504   0627  2      [CLH_WRITE_OUT] :
  505   0628  2          ! Write one record to the output file.
  506   0629  2          BEGIN
  507   0630  3
  508 U 0631  3      %IF FLIP %THEN
  509 UU 0632  3          LOCAL
  510 U  0633  3              temp_record : $flip_rnotxt;
  511   0634  3      %FI
  512   0635  3
  513   0636  3          ! Append carriage control information to the record
  514   0637  3          fs_wchar (fra, %O'15');                  !Carriage return
  515   0638  3          fs_wchar (fra, %O'12');                  !Line feed
  516   0639  3
  517 U 0640  3      %IF FLIP %THEN
  518 U 0641  3          IF  (.gca_op_dev EQL op_dev_flip)
  519 U 0642  3          THEN
  520 U 0643  3              BEGIN
  521 U 0644  3              temp_record [rnotxt_code]   = flip$k_rnotxt;
  522 U 0645  3              temp_record [rnotxt_length] = .fs_length (fra);
  523 U 0646  3              CH$MOVE (.fs_length (fra)
  524 U 0647  3                      ..fs_start (fra)
  525 U 0648  3                      .CH$PTR (temp_record [rnotxt_text]));
  526 U 0649  3              status = $XPO_PUT (IOB    = .rnoiob,
  527 U 0650  3                                 STRING = (flip$k_rnotxt_basesiz +
  528 U 0651  3                                           .fs_length (fra),
  529 U 0652  3                                           CH$PTR (temp_record)));
  530 U 0653  3              END
  531 U 0654  3          ELSE
  532   0655  3      %FI
  533   0656  3
  534 P 0657  3          status = $XPO_PUT  (IOB    = .rnoiob,
  535 P 0658  3                              STRING = (.fs_length (fra),
  536   0659  3                                        .fs_start  (fra)));
  537   0660  3
  538   0661  3          ! Remove the appended characters from the end of the buffer
  539   0662  3          fs_next   (fra) = CH$PLUS (.fs_next (fra), -2);   !Back up pointer
  540   0663  3          fs_length (fra) = .fs_length (fra) - 2;           !Back up counter
  541   0664  3
  542   0665  3          IF  .status
  543   0666  3          THEN
  544   0667  3              RETURN clh_normal;
  545   0668  3
  546   0669  2          END;
  547   0670  2
```

H 15

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 13
V04-000   CLH -- Write 1 record to O/P file w/o CRLF suff 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (9)

```
  549       0671  2  %sbttl 'CLH -- Write 1 record to O/P file w/o CRLF suffix'
  550       0672  2
  551       0673  2            [CLH_OUT_NOCRLF] :
  552       0674  3                ! Write one record to the output file. Don't add carriage control information.
  553       0675  3                BEGIN
  554       0676  3                IF NOT .gca_skip_out
  555       0677  3                THEN
  556       0678  4                    BEGIN
  557  U    0679  4  %IF FLIP %THEN
  558  U    0680  4                LOCAL
  559  U    0681  4                    TEMP_RECORD : $FLIP_RNOTXT;
  560  U    0682  4
  561  U    0683  4                IF  (.gca_op_dev EQL op_dev_flip)
  562  U    0684  4                THEN
  563  U    0685  4                    BEGIN
  564  U    0686  4                    TEMP_RECORD[RNOTXT_CODE] = FLIP$K_RNOTXT;
  565  U    0687  4                    TEMP_RECORD[RNOTXT_LENGTH] = .FS_LENGTH(FRA);
  566  U    0688  4                    CH$MOVE( .FS_LENGTH(FRA),.FS_START(FRA)
  567  U    0689  4                            ,CH$PTR(TEMP_RECORD[RNOTXT_TEXT]));
  568  U    0690  4                    STATUS = $XPO_PUT( IOB=.RNOIOB
  569  U    0691  4                            ,STRING=( FLIP$K_RNOTXT_BASESIZ+.FS_LENGTH(FRA)
  570  U    0692  4                            ,CH$PTR(TEMP_RECORD) ) );
  571  U    0693  4                    END
  572  U    0694  4                ELSE
  573       0695  4  %FI
  574       0696  4
  575  P    0697  4                STATUS = $XPO_PUT ( IOB = .RNOIOB
  576  P    0698  4                            ,STRING = ( .FS_LENGTH (FRA)
  577       0699  4                            ,.FS_START (FRA) ) );
  578       0700  3                END;
  579       0701  3
  580       0702  3                IF  .STATUS OR .gca_skip_out
  581       0703  3                THEN
  582       0704  3                    RETURN CLH_NORMAL;
  583       0705  3
  584       0706  2                END;
  585       0707  2
```

```
 587      0708   2  %sbttl 'CLH -- file closing functions'
 588      0709   2          [CLH_CLOSE_INPUT] :
 589      0710   2              ! Close current input file.
 590      0711   3              BEGIN
 591    P 0712   3              STATUS = $XPO_CLOSE ( IOB = .RNIIOB
 592      0713   3                                ,FAILURE = grab_resultant);
 593      0714   3              RETURN CLH_NORMAL;
 594      0715   2              END;
 595      0716   2
 596      0717   2          [CLH_CLOSE_OUT] :
 597      0718   2              ! Close output file
 598      0719   3              BEGIN
 599    P 0720   3              STATUS = $XPO_CLOSE ( IOB = .RNOIOB
 600      0721   3                                ,FAILURE = grab_resultant);
 601      0722   3              RETURN CLH_NORMAL;
 602      0723   2              END;
 603      0724   2
 604      0725   2          [CLH_CLOSE_DEL_OUT] :
 605      0726   2              ! Close output file
 606      0727   3              BEGIN
 607      0728   3
 608      0729   3              IF  .RNOIOB[IOB$V_TERMINAL]
 609      0730   3              THEN                    ! If it's a terminal,
 610    P 0731   3                  STATUS = $XPO_CLOSE ( IOB = .RNOIOB        !    just close it.
 611      0732   4                                    ,FAILURE = grab_resultant)
 612      0733   3              ELSE
 613      0734   4                  BEGIN
 614    P 0735   4                  STATUS = $XPO_CLOSE ( IOB = .RNOIOB       ! Otherwise, close
 615    P 0736   4                                    ,OPTIONS = REMEMBER !    and delete it.
 616      0737   4                                    ,FAILURE = grab_resultant);
 617    P 0738   4                  STATUS = $XPO_DELETE ( IOB = .RNOIOB
 618      0739   4                                    ,FAILURE = grab_resultant);
 619      0740   3                  END;
 620      0741   3
 621      0742   3              RETURN CLH_NORMAL;
 622      0743   2              END;
 623      0744   2
```

```
:   625    0745  2  %sbttl 'CLH -- Push IOB onto stack'
:   626    0746  2
:   627    0747  2          [CLH_PUSH] :
:   628    0748  3              BEGIN
:   629    0749  3
:   630    0750  3              !  If there are not too many files already open allocate
:   631    0751  3              !  a new IOB on IOBSTK (pointed to by RNIIOB).
:   632    0752  3
:   633    0753  3              IF   .GCA_REQ_DEPTH NEQ .GCA_MAX_REQUIRE
:   634    0754  3              THEN
:   635    0755  4                  BEGIN
:   636    0756  4                  RNIIOB = IOBSTK + (IOB$K_LENGTH * %UPVAL) * .GCA_REQ_DEPTH;
:   637    0757  4                  GCA_REQ_DEPTH = .GCA_REQ_DEPTH + 1;
:   638    0758  4                  RETURN CLH_NORMAL;
:   639    0759  4                  END
:   640    0760  3              ELSE                        ! No room on IOB stack for the IOB.
:   641    0761  3                  RETURN CLH_NO_SPACE;
:   642    0762  3
:   643    0763  2              END;
:   644    0764  2
```

K 15

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 16
V04-000   CLH -- Pop IOB from stack and reaccess previous 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1 (12)

```
646    0765  2    %sbttl 'CLH -- Pop IOB from stack and reaccess previous one'
647    0766  2
648    0767  2    [CLH_POP] :
649    0768  2        ! Cause RUNOFF to stop reading from the current file
650    0769  2        ! and read from the previous file instead.
651    0770  3        BEGIN
652    0771  3
653    0772  3        IF  .GCA_REQ_DEPTH EQL 0
654    0773  3        THEN
655    0774  4            BEGIN
656    0775  4            ! Internal logic error: should not try to
657    0776  4            ! pop the main file.
658    0777  4            0
659    0778  4            END
660    0779  3        ELSE
661    0780  4            BEGIN
662    0781  4            ! Forcefully terminate open .LIST, .NOTE and .IF commands that
663    0782  4            ! are still open when the file in which they occured is about
664    0783  4            ! to be closed.  I.E., do not allow .END commands to be in a
665    0784  4            ! different file than the the opening .LIST, .NOTE,... command.
666    0785  4            TSTTFE (.GCA_REQ_DEPTH);
667    0786  4
668    0787  4            IF  .RNIIOB [IOB$V_OPEN]                    ! Really close the file if
669    0788  4            THEN                                       !  there was a file opened
670    0789  4                ! ( see REQUIR.BLI for case when it's not )
671  P 0790  4                STATUS = $XPO_CLOSE ( IOB = .RNIIOB
672    0791  4                                     ,FAILURE = grab_resultant);
673    0792  4            !Now do pop the file stack.
674    0793  4            RNIIOB = .RNIIOB - (IOB$K_LENGTH * %UPVAL);
675    0794  4            GCA_REQ_DEPTH = .GCA_REQ_DEPTH - 1;
676    0795  4
677    0796  4            IF  .GCA_REQ_DEPTH EQL 0
678    0797  4            THEN                   !Popped all the way back to primary input file
679    0798  4                RNIIOB = .RNEIOB;                  !Get real primary IOB.
680    0799  4
681    0800  4            ! The routine ERROR needs the following information in IRAC.
682    0801  5            BEGIN
683    0802  5            ! Pick of the name and length of the file spec
684    0803  5            BIND
685    0804  5                FILE_SPEC_STUFF = RNIIOB [IOB$T_RESULTANT] : $STR_DESCRIPTOR ();
686    0805  5            IRAC_FSPECP = .FILE_SPEC_STUFF [STR$A_POINTER];
687    0806  5            IRAC_FSPECC = .FILE_SPEC_STUFF [STR$H_LENGTH];
688    0807  4            END;
689    0808  4            IRAC_ISEQN = .RNIIOB [IOB$G_SEQ_NUMB];
690    0809  4            IRAC_IPAGEN = .RNIIOB [IOB$A_PAGE_NUMB];
691    0810  4            RETURN CLH_NORMAL;
692    0811  3            END;
693    0812  3
694    0813  2        END;
695    0814  2
```

```
697        0815  2  %sbttl 'CLH -- Open REQUIRE file spec.'
698        0816  2
699        0817  2          [CLH_OPEN_REQ] :
700        0818  2          ! Open a file requested on a .REQUIRE command.
701        0819  2          BEGIN
702        0820  3          status = $XPO_IOB_INIT (IOB = .rniiob);
703     P  0821  3          status = $XPO_OPEN  (IOB = .rniiob
704     P  0822  3                                  ,OPTIONS = INPUT  !  INPUT file
705     P  0823  3                                  ,DEFAULT = '.RNO' ! Default the extension only!
706     P  0824  3                                  ,FILE_SPEC = ( .fs_length(fs01)      ! filename
707     P  0825  3                                              , .fs_start (fs01))
708        0826  3                                  ,FAILURE = grab_resultant);
709        0827  3
710        0828  3          IF  .status
711        0829  3          THEN
712        0830  3              ! Reset input line/page counters.
713        0831  4              BEGIN
714        0832  4              irac_ipagen = 1;
715        0833  4              irac_iseqn = 1;
716        0834  5              BEGIN
717        0835  5              ! Pick off the name and length of the filespec.
718        0836  5              BIND
719        0837  5                  file_spec_stuff = rniiob [iob$t_resultant] : $STR_DESCRIPTOR ();
720        0838  5
721        0839  5              irac_fspecp = .file_spec_stuff [str$a_pointer];
722        0840  5              irac_fspecc = .file_spec_stuff [str$h_length];
723        0841  5
724        0842  5              ! Output name of .REQUIRED file in .MEM file, if user
725        0843  5              ! said /DEBUG:FILES
726        0844  5
727        0845  5              IF  .gca_debug_fil AND NOT .gca_skip_out
728        0846  5              THEN
729        0847  5
730        0848  5                  ! Yes: User said /DEBUG:FILES and output is being
731        0849  5                  ! generated because the current page is included
732        0850  5                  ! in a /PAGES list.
733        0851  5
734        0852  6                  BEGIN
735        0853  6                  LOCAL
736  U     0854  6  %IF FLIP %THEN
737  U     0855  6                      temp_record : $flip_rnotxt,
738        0856  6  %FI
739        0857  6                      work_area : VECTOR [CH$ALLOCATION (100)],
740        0858  6                      work_count,
741        0859  6                      work_ptr;
742        0860  6
743        0861  6                  work_ptr   = CH$PTR (work_area);
744        0862  6                  work_ptr   = CH$MOVE (10, CH$PTR (UPLIT ('.REQUIRE '")), .work_ptr);
745        0863  6                  work_count = 10;
746        0864  6                  work_ptr   = CH$MOVE (.irac_fspecc, .irac_fspecp, .work_ptr);
747        0865  6                  work_count = .work_count + .irac_fspecc;
748        0866  6                  CH$WCHAR_A (%C'"', work_ptr);
749        0867  6                  CH$WCHAR_A (%O'15', work_ptr);        !Carriage return
750        0868  6                  CH$WCHAR_A (%O'12', work_ptr);        !Line feed
751        0869  6                  work_count = .work_count + 3;
752  U     0870  6  %IF FLIP %THEN
753  U     0871  6                  IF  (.gca_op_dev EQL op_dev_flip)
```

```
 754       U 0872  6                              THEN
 755       U 0873  6                                  BEGIN
 756       U 0874  6                                  temp_record [rnotxt_code]   = flip$k_rnotxt;
 757       U 0875  6                                  temp_record [rnotxt_length] = .work_count;
 758       U 0876  6                                  CH$MOVE (.work_count, CH$PTR (work_area),
 759       U 0877  6                                          CH$PTR (temp_record [rnotxt_text]));
 760       U 0878  6                                  status = $XPO_PUT (IOB = .rnoiob,
 761       U 0879  6                                              STRING = (flip$k_rnotxt_basesiz +
 762       U 0880  6                                                        .work_count,
 763       U 0881  6                                                        CH$PTR(temp_record)));
 764       U 0882  6                                  END
 765       U 0883  6                              ELSE
 766         0884  6 %FI
 767       P 0885  6                                  $XPO_PUT (IOB = .rnoiob,
 768         0886  6                                          STRING = (.work_count, CH$PTR (work_area)));
 769         0887  5                              END;
 770         0888  4                          END;
 771         0889  4
 772         0890  4                      RETURN clh_normal;
 773         0891  4
 774         0892  4                      END
 775         0893  3                  ELSE
 776         0894  3                      RETURN clh_cant_open;
 777         0895  3
 778         0896  2                  END;
 779         0897  2
```

N 15

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 19
V04-000   CLH -- Close REQUIRE file (NULL) and bad OPCODE 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (14)

```
781    0898  2  %sbttl 'CLH -- Close REQUIRE file (NULL) and bad OPCODE processing'
782    0899  2
783    0900  2      [CLH_CLOSE_REQ] :
784    0901  3          ! Close a ".REQUIRE" file.
785    0902  3          BEGIN
786    0903  3          0
787    0904  2          END;
788    0905  2
```

B 16

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 20
V04-000   Open initialization file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNGFF.SRC]CLH.BLI;1   (15)

```
 790      0906   2 %SBTTL 'Open initialization file'
 791      0907   2
 792      0908   2            [CLH_OPEN_INIT] :
 793      0909   2
 794      0910   2                ! Open initialization file specified in fs01.
 795      0911   2
 796      0912   3                BEGIN
 797      0913   3                status = $XPO_IOB_INIT (IOB   = .rniiob);
 798    P 0914   3                status = $XPO_OPEN
 799    P 0915   3                        (IOB     = .rniiob,
 800    P 0916   3                         OPTIONS = input,
 801    P 0917   3                         FILE_SPEC = (.fs_length (fs01),
 802    P 0918   3                                      .fs_start (fs01)),
 803      0919   3                         FAILURE = grab_resultant);
 804      0920   3                IF  .status
 805      0921   3                THEN
 806      0922   3                    !
 807      0923   3                    ! Succeeded in opening file.
 808      0924   3                    !
 809      0925   4                    BEGIN
 810      0926   4                    !
 811      0927   4                    ! Pick off name and length of filespec.
 812      0928   4                    !
 813      0929   4                    BIND
 814      0930   4                        file_spec_stuff = rniiob [iob$t_resultant] : $STR_DESCRIPTOR ();
 815      0931   4
 816      0932   4                    irac_fspecp = .file_spec_stuff [str$a_pointer];
 817      0933   4                    irac_fspecc = .file_spec_stuff [str$h_length];
 818      0934   4
 819      0935   4                    ! Output name of initialization file in .MEM file, if user
 820      0936   4                    ! said /DEBUG:FILES
 821      0937   4
 822      0938   4                    IF  .gca_debug_fil AND NOT .gca_skip_out
 823      0939   4                    THEN
 824      0940   4                        !
 825      0941   4                        ! Yes, user said /DEBUG:FILES and output is being
 826      0942   4                        ! generated because the current page is included
 827      0943   4                        ! in a /PAGES list.
 828      0944   4                        !
 829      0945   5                        BEGIN
 830      0946   5                        LOCAL
 831      0947   5
 832    U 0948   5 %IF FLIP %THEN
 833      0949   5                            temp_record : $flip_rnotxt,
 834      0950   5 %FI
 835      0951   5                            work_area : VECTOR [CH$ALLOCATION (100)],
 836      0952   5                            work_count,
 837      0953   5                            work_ptr;
 838      0954   5
 839      0955   5                        work_ptr = CH$PTR (work_area);
 840      0956   5
 841      0957   5                        ! Move descriptive text into work area.
 842      0958   5                        ! Identifies DSR$INIT if only 8 characters long.
 843      0959   5                        ! Otherwise, it must be for DSRPLUS$INIT.
 844      0960   5
 845      0961   5                        IF  .fs_length (fs01) EQL 8
 846      0962   5                        THEN
```

```
 847          0963  6                                              BEGIN
 848          0964  6                                              work_count = 15;
 849          0965  6                                              work_ptr   = CH$MOVE (.work_count,
 850          0966  6                                                                    CH$PTR (UPLIT ('DSR$INIT file '"')),
 851          0967  6                                                                    .work_ptr);
 852          0968  6                                              END
 853          0969  5                                      ELSE
 854          0970  6                                              BEGIN
 855          0971  6                                              work_count = 19;
 856          0972  6                                              work_ptr   = CH$MOVE (.work_count,
 857          0973  6                                                                    CH$PTR (UPLIT ('DSRPLUS$INIT file '"')),
 858          0974  6                                                                    .work_ptr);
 859          0975  5                                              END;

 860          0976  5                                      !
 861          0977  5                                      ! Add file name to work area.
 862          0978  5                                      !
 863          0979  5                                      work_ptr   = CH$MOVE (.irac_fspecc, .irac_fspecp, .work_ptr);
 864          0980  5                                      work_count = .work_count + .irac_fspecc;

 865          0981  5                                      !
 866          0982  5                                      ! Add end-of-line characters to work area.
 867          0983  5                                      !
 868          0984  5                                      CH$WCHAR_A (%C'"',  work_ptr);
 869          0985  5                                      CH$WCHAR_A (%O'15', work_ptr);           !Carriage return
 870          0986  5                                      CH$WCHAR_A (%O'12', work_ptr);           !Line feed
 871          0987  5                                      work_count = .work_count + 3;
 872        U 0988  5   %IF FLIP %THEN
 873        U 0989  5                                      ! if FLIP, send initialization file info to user
 874        U 0990  5                                      ! in correct form for FLIP.
 875        U 0991  5                                      !
 876        U 0992  5                                      IF  (.gca_op_dev EQL op_dev_flip)
 877        U 0993  5                                      THEN
 878        U 0994  5                                              BEGIN
 879        U 0995  5                                              temp_record [rnotxt_code]   = flip$k_rnotxt;
 880        U 0996  5                                              temp_record [rnotxt_length] = .work_count;
 881        U 0997  5                                              CH$MOVE (.work_count, CH$PTR (work_area),
 882        U 0998  5                                                       CH$PTR (temp_record [rnotxt_text]));
 883        U 0999  5                                              status = $XPO_PUT (IOB    = .rnoiob,
 884        U 1000  5                                                                 STRING = (flip$k_rnotxt_basesiz +
 885        U 1001  5                                                                           .work_count,
 886        U 1002  5                                                                           CH$PTR(temp_record)));
 887        U 1003  5                                              END
 888        U 1004  5                                      ELSE
 889          1005  5   %FI
 890          1006  5
 891          1007  5                                      ! Otherwise, send initialization file info to user
 892          1008  5                                      ! in DSRPLUS form.
 893          1009  5                                      !
 894        P 1010  5                                      $XPO_PUT (IOB    = .rnoiob,
 895          1011  5                                                STRING = (.work_count, CH$PTR (work_area)));
 896          1012  4                                      END;

 897          1013  4
 898          1014  4                              RETURN clh_normal;
 899          1015  4                              END
 900          1016  3                      ELSE
 901          1017  3                              !
 902          1018  3                              ! Couldn't find file to open.
 903          1019  3                              !
```

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16 VAX-11 Bliss-32 V4.0-742   Page 22
V04-000   Open initialization file   14-Sep-1984 13:05:41 DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1 (15)

D 16

```
 904    1020  3                      RETURN clh_cant_open;
 905    1021  3
 906    1022  3              END;
 907    1023  3
 908    1024  2              [OUTRANGE] :
 909    1025  2                  ! Error in program.
 910    1026  2                  BEGIN
 911    1027  3                  PUTMSG (rnfile, CH$PTR (UPLIT ('CLH')), 3);
 912    1028  3                  RETURN clh_end_file;              ! Make a guess
 913    1029  3                  END;
 914    1030  2
 915    1031  2          TES;
 916    1032  2          0
 917    1033  2
 918    1034  1      END;                                          ! End of CLH
```

```
                                        .TITLE  CLH file processing interface and command line
;                                               hand
;                                       .IDENT  \V04-000\

                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    4F  4E  52  2E  00000 P.AAE:   .ASCII  \.RNO\
                    4F  4E  52  2E  00004 P.AAF:   .ASCII  \.RNO\
        00 00 22 20 45 52 49 55 51 45 52 2E 00008 P.AAG:   .ASCII  \.REQUIRE "\<0><0>
22 20 65 6C 69 66 20 54 49 4E 49 24 52 53 44 00014 P.AAH:   .ASCII  \DSR$INIT file "\<0>
                                    00 00023
69 66 20 54 49 4E 49 24 53 55 4C 50 52 53 44 00024 P.AAI:   .ASCII  \DSRPLUS$INIT file "\<0>
                                00 22 20 65 6C 00033
                                    00 48 4C 43 00038 P.AAJ:   .ASCII  \CLH\<0>

                                        .PSECT  $OWN$,NOEXE,2

                            00000 STATUS: .BLKB  4
                            00004 DEF_OUT_LNG:
                                            .BLKB  4
                            00008 DEF_OUT_SPC:
                                            .BLKB  52
                     0004   0003C $IOB$DEFAULT:
                                            .WORD  4
                     01 0E  0003E           .BYTE  14, 1
             00000000' 00040                .ADDRESS P.AAF

                                        .EXTRN  FRA, GCA, FS01, IRA
                                        .EXTRN  IRAC, IPFTYP, IPFTOP
                                        .EXTRN  OPFTOP, IOBSTK, RNEIOB
                                        .EXTRN  RNIIOB, RNOIOB, TSIIOB
                                        .EXTRN  TTIIOB, TTOIOB, IPFTCT
                                        .EXTRN  RNFILE, RNFRTL, ERMS
                                        .EXTRN  GRAB RESULTANT, PUTMSG
                                        .EXTRN  TSTTFE, XST$FORMAT
                                        .EXTRN  XPO$OPEN, XPO$GET
                                        .EXTRN  XPO$FAILURE, XPO$PUT
                                        .EXTRN  XPO$CLOSE, XPO$DELETE

                                        .PSECT  $CODE$,NOWRT,2
```

```
                                        E 16
CLH          file processing interface and command line hand 15-Sep-1984 23:56:16    VAX-11 Bliss-32 V4.0-742        Page  23
V04-000      Open initialization file                        14-Sep-1984 13:05:41    DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1  (15)
```

```
                                        0FFC  00000                .ENTRY   CLH, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 0289
                            5B 00000000G EF 9E  00002                MOVAB    IRA+4, R11
                            5A 00000000G EF 9E  00009                MOVAB    GCA+188, R10
                            59 00000000G EF 9E  00010                MOVAB    RNIIOB, R9
                            58 00000000G EF 9E  00017                MOVAB    IRAC+20, R8
                            57 00000000' EF 9E  0001E                MOVAB    STATUS, R7
                            5E       94 AE 9E  00025                MOVAB    -108(SP), SP
                            01       04 AC CF  00029                CASEL    OPCODE, #1, #12                             ; 0318
     02F3        02EE       0089        0032  0002E 1$:   .WORD    3$-1$,-
     0388        036A       022F        013A  00036                4$-1$,-
     0311        02A5       0578        03E6  0003E                29$-1$,-
                                        0488  00046                30$-1$,-
                                                                  11$-1$,-
                                                                  24$-1$,-
                                                                  37$-1$,-
                                                                  39$-1$,-
                                                                  44$-1$,-
                                                                  54$-1$,-
                                                                  26$-1$,-
                                                                  33$-1$,-
                                                                  48$-1$
                                        03    DD  00048                PUSHL    #3                                      ; 1027
                            00000000' EF 9F  0004A                PUSHAB   P.AAJ
                            00000000G 8F DD  00050                PUSHL    #RNFILE
                            00000000G EF 03 FB  00056                CALLS    #3, PUTMSG
                                        0542  31 0005D 2$:   BRW      53$                                               ; 1028
                                        52    69 D0  00060 3$:   MOVL     RNIIOB, R2                                    ; 0354
                            64 AE       04 B0  00063                MOVW     #4, $STR$STRING
                            66 AE       0E 90  00067                MOVB     #14, $STR$STRING+2
                            67 AE       01 90  0006B                MOVB     #1, $STR$STRING+3
                            68 AE 00000000' EF 9E  0006F                MOVAB    P.AAE, $STR$STRING+4
                                        7E    D4  00077                CLRL     -(SP)
                                  68 AE 9F  00079                PUSHAB   $STR$STRING
                                        7E    D4  0007C                CLRL     -(SP)
                            00000000G EF 03 FB  0007E                CALLS    #3, XST$FORMAT
                            08 A2       50 D0  00085                MOVL     R0, 8(R2)
                            2E A2       01 88  00089                BISB2    #1, 46(R2)
                            2C A2       01 90  0008D                MOVB     #1, 44(R2)
                            00000000G EF 9F  00091                PUSHAB   GRAB_RESULTANT
                                        7E    D4  00097                CLRL     -(SP)
                                        52    DD  00099                PUSHL    R2
                            00000000G EF 03 FB  0009B                CALLS    #3, XPO$OPEN
                                        67    50 D0  000A2                MOVL     R0, STATUS
                                        B5    67 E9  000A5                BLBC     STATUS, 2$                           ; 0356
                            50          1C C1  000A8                ADDL3    #28, RNIIOB, R0                            ; 0360
                            FC A8       04 A0 D0  000AC                MOVL     4(R0), IRAC+16                           ; 0364
                                  68    3C A0  000B1                MOVZWL   (R0), IRAC+20                              ; 0365
                                        04E7  31 000B4                BRW      52$                                      ; 0376
                            7E          69 1C C1  000B7 4$:   ADDL3    #28, RNIIOB, -(SP)                               ; 0383
                            00000000V EF 01 FB  000BB                CALLS    #1, GET_OUT_DEFAULT
     53        14 AA       04          04 EF  000C2                EXTZV    #4, #4, -GCA+208, R3                        ; 0385
               05          00          53 CF  000C8                CASEL    R3, #0, #5
     0093        0093       000F        000F  000CC 5$:   .WORD    6$-5$,-
                            0048        0048  000D4                6$-5$,-
                                                                  9$-5$,-
                                                                  9$-5$,-
```

CLH
V04-000
F 16
file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 24
Open initialization file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)

```
                                                              7$-5$,-
                                                              7$-5$
                              0084  31  000D8           BRW    9$
                 52 00000000G EF D0 000DB  6$:  MOVL   RNOIOB, R2
              64 AE     04    A7 B0 000E2       MOVW   DEF_OUT_LNG, $STR$STRING      0401
              66 AE           0E 90 000E7       MOVB   #14, $STR$STRING+2
              67 AE           01 90 000EB       MOVB   #1, $STR$STRING+3
              68 AE     08    A7 9E 000EF       MOVAB  DEF_OUT_SPC, $STR$STRING+4
                              7E D4 000F4       CLRL   -(SP)
                           68 AE 9F 000F6       PUSHAB $STR$STRING
                              7E D4 000F9       CLRL   -(SP)
              00000000G EF    03 FB 000FB       CALLS  #3, XST$FORMAT
           51          69     1C C1 00102       ADDL3  #28, RNIIOB, R1
              08 A2           50 7D 00106       MOVQ   R0, 8(R2)
              2E A2  00040002 8F C8 0010A       BISL2  #262146, 46(R2)
                              33 11 00112       BRB    8$
                 52 00000000G EF DC 00114  7$:  MOVL   RNOIOB, R2                    0444
              64 AE     04    A7 B0 0011B       MOVW   DEF_OUT_LNG, $STR$STRING
              66 AE           0E 90 00120       MOVB   #14, $STR$STRING+2
              67 AE           01 90 00124       MOVB   #1, $STR$STRING+3
              68 AE     08    A7 9E 00128       MOVAB  DEF_OUT_SPC, $STR$STRING+4
                              7E D4 0012D       CLRL   -(SP)
                           68 AE 9F 0012F       PUSHAB $STR$STRING
                              7E D4 00132       CLRL   -(SP)
           51 00000000G EF    03 FB 00134       CALLS  #3, XST$FORMAT
                       69     1C C1 0013B       ADDL3  #28, RNIIOB, R1
              08 A2           50 7D 0013F       MOVQ   R0, 8(R2)
              2E A2           02 88 00143       BISB2  #2, 46(R2)
              2C A2           01 90 00147  8$:  MOVB   #1, 44(R2)
              00000000G EF    9F 0014B          PUSHAB GRAB_RESULTANT
                              7E D4 00151       CLRL   -(SP)
                              52 DD 00153       PUSHL  R2
              00000000G EF    03 FB 00155       CALLS  #3, XPO$OPEN
                       67     50 D0 0015C       MOVL   R0, STATUS
                       03     67 E8 0015F  9$:  BLBS   STATUS, 10$                   0465
                            043D 31 00162       BRW    53$
                            0436 31 00165 10$:  BRW    52$
                       50     69 D0 00168 11$:  MOVL   RNIIOB, R0                    0477
                    2C A0     06 90 0016B       MOVB   #6, 44(R0)
              00000000G EF    9F 0016F          PUSHAB XPO$FAILURE
                              7E D4 00175       CLRL   -(SP)
                              50 DD 00177       PUSHL  R0
              00000000G EF    03 FB 00179       CALLS  #3, XPO$GET
                       67     50 D0 00180       MOVL   R0, STATUS
                       51     67 D0 00183       MOVL   STATUS, R1                    0479
                       0C     51 E8 00186       BLBS   R1, 12$
              0020A22A 8F     51 D1 00189       CMPL   R1, #2138666                  0480
                              03 13 00190       BEQL   12$
                            00A0 31 00192       BRW    20$
                       50     69 D0 00195 12$:  MOVL   RNIIOB, R0                    0485
           F4 A8     4C A0    D0 00198          MOVL   76(R0), IRAC+8
           F8 A8     4A A0    32 0019D          CVTWL  74(R0), IRAC+12               0486
    F0 A8  30 A0  01 04 EF 001A2               EXTZV  #4, #1, 48(R0), IRAC+4         0487
              0020A22A 8F     51 D1 001A9       CMPL   R1, #2138666                  0490
                              1C 12 001B0       BNEQ   14$
                           7E 34 A0 3C 001B2    MOVZWL 52(R0), -(SP)
                       32     6E B1 001B6       CMPW   (SP), #50                     0494
```

```
G 16
CLH                 file processing interface and command line hand 15-Sep-1984 23:56:16     VAX-11 Bliss-32 V4.0-742          Page 25
V04-000             Open initialization file                        14-Sep-1984 13:05:41     DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)
```

```
                                       03 1B 001B9          BLEQU   13$
                          6E           32 D0 001BB          MOVL    #50, (SP)
                                 38    A0 DD 001BE  13$:     PUSHL   56(R0)                               0493
                  00000000G      00000000G 8F DD 001C1      PUSHL   #RNFRTL                               0492
         00000000G EF            03 FB 001C7              CALLS   #3, ERMS
                          51           69 00 001CE  14$:     MOVL    RNIIOB, R1                           0498
                    FC AB         38    A1 D0 001D1          MOVL    56(R1), IRA                          0499
                       6B         FC    AB D0 001D6          MOVL    IRA, IRA+4
                    04 AB         34    A1 3C 001DA          MOVZWL  52(R1), IRA+8                         0500
                    08 AB         04    AB D0 001DF          MOVL    IRA+8, IRA+12                         0501
                                  08    AB D5 001E4  15$:     TSTL    IRA+12                              0521
                                        49 15 001E7          BLEQ    19$
                          54           6B D0 001E9          MOVL    IRA+4, PTR                             0525
                          50           64 9A 001EC          MOVZBL  (PTR), X                               0530
                          52           50 D0 001EF          MOVL    X, R2                                  0532
                          53           01 D0 001F2          MOVL    #1, R3
                          0C           52 D1 001F5          CMPL    R2, #12                               C535
                                        18 12 001F8          BNEQ    16$
                                        53 D4 001FA          CLRL    R3
            11        30 A1           04 E0 001FC          BBS     #4, 48(R1), 16$                        0551
                             F8    A8 D6 00201          INCL    IRAC+12                                0554
                    F4 A8           01 D0 00204          MOVL    #1, IRAC+8                             0555
                    4C A1      F4    A8 D0 00208          MOVL    IRAC+8, 76(R1)                         0556
                    4A A1      F8    A8 B0 0020D          MOVW    IRAC+12, 74(R1)                        0557
                                        52 D5 00212  16$:     TSTL    R2                                  0562
                                        0E 13 00214          BEQL    17$
                          0C           52 D1 00216          CMPL    R2, #12
                                        09 13 00219          BEQL    17$
            0000007F 8F           52 D1 0021B          CMPL    R2, #127
                                        0B 12 00222          BNEQ    18$
                                        53 D4 00224  17$:     CLRL    R3
                          50     00    BB 9A 00226          MOVZBL  @IRA+4, X                              0567
                                       6B D6 0022A          INCL    IRA+4
                                 08    AB D7 0022C          DECL    IRA+12
                          B2           53 E9 0022F  18$:     BLBC    R3, 15$                              0570
                                    0369 31 00232  19$:     BRW     52$                                  0583
            00209000 8F           51 D1 00235  20$:     CMPL    R1, #2134016                         0585
                                        18 12 0023C          BNEQ    22$
                                        6A D5 0023E          TSTL    GCA+188                              0589
                                        0F 13 00240          BEQL    21$
                                        08 DD 00242          PUSHL   #8                                   0593
                 FDB7 CF           01 FB 00244          CALLS   #1, CLH
                                        05 DD 00249          PUSHL   #5                                   0607
                 FDB0 CF           01 FB 0024B          CALLS   #1, CLH
                                        04 00250          RET
                          50           02 D0 00251  21$:     MOVL    #2, R0                               0616
                                        03 11 00254          BRB     23$
                          50           03 D0 00256  22$:     MOVL    #3, R0                               0618
                                 08    AB D4 00259  23$:     CLRL    IRA+12                              0611
                                        04 0025C          RET                                            0583
         00000000G FF           0D 90 0025D  24$:     MOVB    #13, @FRA+4                           0637
                  00000000G EF           D6 00264          INCL    FRA+4
                  00000000G EF           D6 0026A          INCL    FRA+12
         00000000G FF           0A 90 00270          MOVB    #10, @FRA+4                           0638
                  00000000G EF           D6 00277          INCL    FRA+4
                  00000000G EF           D6 0027D          INCL    FRA+12
            50 00000000G EF           D0 00283          MOVL    RNOIOB, R0                           0659
```

CLH
V04-000

H 16
file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 26
Open initialization file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)

```
            64  AE  00000000G  EF  B0  0028A          MOVW    FRA+12, $IOB$OUTPUT
            66  AE             0E  90  00292          MOVB    #14, $IOB$OUTPUT+2
            67  AE             01  90  00296          MOVB    #1, $IOB$OUTPUT+3
            68  AE  00000000G  EF  D0  0029A          MOVL    FRA, $IOB$OUTPUT+4
            44  A0         64  AE  9E  002A2          MOVAB   $IOB$OUTPUT, 68(R0)
            2C  A0             07  90  002A7          MOVB    #7, 44(R0)
                    00000000G  EF  9F  002AB          PUSHAB  XPO$FAILURE
                               7E  D4  002B1          CLRL    -(SP)
                               50  DD  002B3          PUSHL   R0
        00000000G  EF          03  FB  002B5          CALLS   #3, XPO$PUT
                    67         50  D0  002BC          MOVL    R0, STATUS
        00000000G  EF          02  C2  002BF          SUBL2   #2, FRA+4                              0662
        00000000G  EF          02  C2  002C6          SUBL2   #2, FRA+12                             0663
                    4A         67  E8  002CD          BLBS    STATUS, 28$                            0665
                       02D3    31  002D0  25$:        BRW     54$
                           3C  B4  AA  E8  002D3  26$:  BLBS  GCA+112, 27$                           0676
                    50  00000000G  EF  D0  002D7      MOVL    RNOIOB, R0                             0699
            64  AE  00000000G  EF  B0  002DE          MOVW    FRA+12, $IOB$OUTPUT
            66  AE             0E  90  002E6          MOVB    #14, $IOB$OUTPUT+2
            67  AE             01  90  002EA          MOVB    #1, $IOB$OUTPUT+3
            68  AE  00000000G  EF  D0  002EE          MOVL    FRA, $IOB$OUTPUT+4
            44  A0         64  AE  9E  002F6          MOVAB   $IOB$OUTPUT, 68(R0)
            2C  A0             07  90  002FB          MOVB    #7, 44(R0)
                    00000000G  EF  9F  002FF          PUSHAB  XPO$FAILURE
                               7E  D4  00305          CLRL    -(SP)
                               50  DD  00307          PUSHL   R0
        00000000G  EF          03  FB  00309          CALLS   #3, XPO$PUT
                    67         50  D0  00310          MOVL    R0, STATUS
                    04         67  E8  00313  27$:     BLBS    STATUS, 28$                           0702
                    B6     B4  AA  E9  00316          BLBC    GCA+112, 25$
                               7A  11  0031A  28$:     BRB     36$                                   0704
                    50         69  D0  0031C  29$:     MOVL    RNIIOB, R0                            0713
                               07  11  0031F          BRB     31$
                    50  00000000G  EF  D0  00321  30$:  MOVL   RNOIOB, R0                            0721
            2C  A0             02  90  00328  31$:     MOVB    #2, 44(R0)
                    00000000G  EF  9F  0032C          PUSHAB  GRAB_RESULTANT
                               7E  D4  00332          CLRL    -(SP)
                               50  DD  00334          PUSHL   R0
        00000000G  EF          03  FB  00336  32$:     CALLS   #3, XPO$CLOSE
                               54  11  0033D          BRB     35$
                    52  00000000G  EF  D0  0033F  33$:  MOVL   RNOIOB, R2                            0729
        10          32  A2     04  E1  00346          BBC     #4, 50(R2), 34$
            2C  A2             02  90  0034B          MOVB    #2, 44(R2)                             0732
                    00000000G  EF  9F  0034F          PUSHAB  GRAB_RESULTANT
                               7E  D4  00355          CLRL    -(SP)
                    52         52  DD  00357          PUSHL   R2
                               DB  11  00359          BRB     32$
            2E  A2             10  88  0035B  34$:     BISB2   #16, 46(R2)                           0737
            2C  A2             02  90  0035F          MOVB    #2, 44(R2)
                    00000000G  EF  9F  00363          PUSHAB  GRAB_RESULTANT
                               7E  D4  00369          CLRL    -(SP)
                    52         52  DD  0036B          PUSHL   R2
        00000000G  EF          03  FB  0036D          CALLS   #3, XPO$CLOSE
                    67         50  D0  00374          MOVL    R0, STATUS
                    50  00000000G  EF  D0  00377      MOVL    RNOIOB, R0                             0739
            2C  A0             03  90  0037E          MOVB    #3, 44(R0)
                    00000000G  EF  9F  00382          PUSHAB  GRAB_RESULTANT
```

CLH   I 16   Page 27
V04-000   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742
           Open initialization file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)

```
                                       7E  D4 00388           CLRL    -(SP)
                                       50  DD 0038A           PUSHL   R0
                        00000000G  EF  03  FB 0038C           CALLS   #3, XPO$DELETE
                                   67  50  D0 00393  35$:     MOVL    R0, STATUS
                                       79  11 00396  36$:     BRB     43$
                    FC  AA           6A  D1 00398  37$:     CMPL    GCA+188, GCA+184
                                       14  13 0039C           BEQL    38$
              50    6A  8F 000000F4  C5 0039E           MULL3   #244, GCA+188, R0
                    69 00000000GEF40 9E 003A6           MOVAB   IOB$TK[R0], RNIIOB
                                       6A  D6 003AE           INCL    GCA+188
                                       5F  11 003B0           BRB     43$
                                   50  05  D0 003B2  38$:     MOVL    #5, R0
                                       04 003B5           RET
                                   50  6A  D0 003B6  39$:     MOVL    GCA+188, R0
                                       03  12 003B9           BNEQ    40$
                                   01E8 31 003BB           BRW     54$
                                   50  DD 003BE  40$:     PUSHL   R0
                        00000000G  EF  01  FB 003C0           CALLS   #1, TSTTFE
                                   50  69  D0 003C7           MOVL    RNIIOB, R0
                    18         32  A0  E9 003CA           BLBC    50(R0), 41$
                    2C  A0           02  90 003CE           MOVB    #2, 44(R0)
                        00000000G  EF  9F 003D2           PUSHAB  GRAB_RESULTANT
                                       7E  D4 003D8           CLRL    -(SP)
                                       50  DD 003DA           PUSHL   R0
                        00000000G  EF  03  FB 003DC           CALLS   #3, XPO$CLOSE
                                   67  50  D0 003E3           MOVL    R0, STATUS
                    69 000000F4  8F  C2 003E6  41$:     SUBL2   #244, RNIIOB
                                       6A  D7 003ED           DECL    GCA+188
                                       07  12 003EF           BNEQ    42$
                    69 00000000G  EF  D0 003F1           MOVL    RNEIOB, RNIIOB
                                   50  69  D0 003F8  42$:     MOVL    RNIIOB, R0
                    51         1C  A0  9E 003FB           MOVAB   28(R0), R1
                FC  A8         04  A1  D0 003FF           MOVL    4(R1), IRAC+16
                    68           61  3C 00404           MOVZWL  (R1), IRAC+20
                F4  A8         4C  A0  D0 00407           MOVL    76(R0), IRAC+8
                F8  A8         4A  A0  32 0040C           CVTWL   74(R0), IRAC+12
                                   018A 31 00411  43$:     BRW     52$
                                   69  56  D0 00414  44$:     MOVL    RNIIOB, R6
        00F4  8F          00    6E  00  2C 00417           MOVC5   #0, (SP), #0, #244, (R6)
                                       66     0041E
                    66 0301003D  8F  D0 0041F           MOVL    #50397245, (R6)
                    1E  A6     020E  8F  B0 00426           MOVW    #526, 30(R6)
                    67 00208001  8F  D0 0042C           MOVL    #2129921, STATUS
                    64  AE 00000000G  EF  B0 00433           MOVW    FS01+12, $STR$STRING
                    66  AE         0E  90 0043B           MOVB    #14, $STR$STRING+2
                    67  AE         01  90 0043F           MOVB    #1, $STR$STRING+3
                    68  AE 00000000G  EF  D0 00443           MOVL    FS01, $STR$STRING+4
                                       7E  D4 0044B           CLRL    -(SP)
                    68  AE  9F 0044D           PUSHAB  $STR$STRING
                                       7E  D4 00450           CLRL    -(SP)
                        00000000G  EF  03  FB 00452           CALLS   #3, XST$FORMAT
                    50  A6         04  D0 00459           MOVL    R0, 4(R6)
                    08  A6     3C  A7  9E 0045D           MOVAB   $IOB$DEFAULT, 8(R6)
                    2E  A6         01  88 00462           BISB2   #1, 46(R6)
                    2C  A6         01  90 00466           MOVB    #1, 44(R6)
                        00000000G  EF  9F 0046A           PUSHAB  GRAB_RESULTANT
                                       7E  D4 00470           CLRL    -(SP)
```

0742
0753
0756
0757
0761
0772
0785
0787
0791
0793
0794
0796
0798
0804
0805
0806
0808
0809
0810
0820
0826

```
                                          J 16
CLH            file processing interface and command line hand 15-Sep-1984 23:56:16     VAX-11 Bliss-32 V4.0-742          Page 28
V04-000        Open initialization file                        14-Sep-1984 13:05:41     DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)
```

```
                       00000000G  EF         56  DD 00472             PUSHL   R6
                                  03         FB    00474             CALLS   #3, XPO$OPEN
                                  67         50  D0 0047B             MOVL    R0, STATUS
                                  03         67  E8 0047E             BLBS    STATUS, 45$                              0828
                                           011E  31 00481             BRW     53$
                             F8   A8         01  D0 00484  45$:       MOVL    #1, IRAC+12                              0832
                             F4   A8         01  D0 00488             MOVL    #1, IRAC+8                               0833
                       50         69         1C  C1 0048C             ADDL3   #28, RNIIOB, R0                          0837
                             FC   A8     04  A0  D0 00490             MOVL    4(R0), IRAC+16                           0839
                                  68         60  3C 00495             MOVZWL  (R0), IRAC+20                            0840
                       03    B8   AA         03  E0 00498             BBS     #3, GCA+116, 47$                         0845
                                           00FE  31 0049D  46$:       BRW     52$
                             F9        B4  AA  E8 004A0  47$:         BLBS    GCA+112, 46$
                             53        08  AE  9E 004A4             MOVAB   WORK_AREA, WORK_PTR                       0861
                       63 00000000'  EF         0A  28 004A8             MOVC3   #10, -P.AAG, (WORK_PTR)                  0862
                                  56         0A  D0 004B0             MOVL    #10, WORK_COUNT                          0863
                                           00A5  31 004B3             BRW     51$                                     0864
                                  56         69  D0 004B6  48$:       MOVL    RNIIOB, R6                               0913
       00F4  8F      00         6E         00  2C 004B9             MOVC5   #0, (SP), #0, #244, (R6)
                                           66     004C0
                       66 0301003D  8F      D0 004C1             MOVL    #50397245, (R6)
                             1E   A6     020E  8F  B0 004C8             MOVW    #526, 30(R6)
                                  67 00208001  8F  D0 004CE             MOVL    #2129921, STATUS
                             64   AE 00000000G  EF  B0 004D5             MOVW    FS01+12, $STR$STRING                      0919
                             66   AE         0E  90 004DD             MOVB    #14, $STR$STRING+2
                             67   AE         01  90 004E1             MOVB    #1, $STR$STRING+3
                             68   AE 00000000G  EF  D0 004E5             MOVL    FS01, $STR$STRING+4
                                  7E         D4 004ED             CLRL    -(SP)
                             68   AE         9F 004EF             PUSHAB  $STR$STRING
                                  7E         D4 004F2             CLRL    -(SP)
                       00000000G  EF         03  FB 004F4             CALLS   #3, XST$FORMAT
                             04   A6         50  D0 004FB             MOVL    R0, 4(R6)
                             2E   A6         01  88 004FF             BISB2   #1, 46(R6)
                             2C   A6         01  90 00503             MOVB    #1, 44(R6)
                       00000000G  EF         9F 00507             PUSHAB  GRAB_RESULTANT
                                  7E         D4 0050D             CLRL    -(SP)
                                  56         DD 0050F             PUSHL   R6
                       00000000G  EF         03  FB 00511             CALLS   #3, XPO$OPEN
                                  67         50  D0 00518             MOVL    R0, STATUS
                                  03         67  E8 0051B             BLBS    STATUS, 49$                              0920
                                           0081  31 0051E             BRW     53$
                       50         69         1C  C1 00521  49$:       ADDL3   #28, RNIIOB, R0                          0930
                             FC   A8     04  A0  D0 00525             MOVL    4(R0), IRAC+16                           0932
                                  68         60  3C 0052A             MOVZWL  (R0), IRAC+20                            0933
                       6C    B8   AA         03  E1 0052D             BBC     #3, GCA+116, 52$                         0938
                             68        B4  AA  E8 00532             BLBS    GCA+112, 52$
                             53        08  AE  9E 00536             MOVAB   WORK_AREA, WORK_PTR                       0955
                             08 00000000G  EF  D1 0053A             CMPL    FS01+12, #8                               0961
                                  0D         12 00541             BNEQ    50$
                                  56         0F  D0 00543             MOVL    #15, WORK_COUNT                          0964
                       63 00000000'  EF      56  28 00546             MOVC3   WORK_COUNT, P.AAH, (WORK_PTR)             0967
                                  0B         11 0054E             BRB     51$                                     0961
                                  56         13  D0 00550  50$:       MOVL    #19, WORK_COUNT                          0971
                       63 00000000'  EF      56  28 00553             MOVC3   WORK_COUNT, P.AAI, (WORK_PTR)             0974
                       63    FC   B8         68  28 0055B  51$:       MOVC3   IRAC+20, @IRAC+16, (WORK_PTR)             0979
                                  56         68  C0 00560             ADDL2   IRAC+20, WORK_COUNT                      0980
                                  83      0D22  8F  B0 00563             MOVW    #3362, (WORK_PTR)+                        0984
```

K 16

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 29
V04-000   Open initialization file   14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (15)

```
                          83              0A  90  00568              MOVB    #10, (WORK_PTR)+                     : 0986
                          56              03  C0  0056B              ADDL2   #3, WORK_COUNT                      : 0987
                          50  00000000G   EF  D0  0056E              MOVL    RN0IOB, R0                          : 1011
                          6E              56  B0  00575              MOVW    WORK_COUNT, $IOB$OUTPUT             :
                    02    AE              0E  90  00578              MOVB    #14, $IOB$OUTPUT+2                  :
                    03    AE              01  90  0057C              MOVB    #1, $IOB$OUTPUT+3                   :
                    04    AE          08  AE  9E  00580              MOVAB   WORK_AREA, $IOB$OUTPUT+4           :
                    44    A0              6E  9E  00585              MOVAB   $IOB$OUTPUT, 68(R0)                :
                    2C    A0              07  90  00589              MOVB    #7, 44(R0)                         :
                        00000000G        EF  9F  0058D              PUSHAB  XPO$FAILURE                        :
                                          7E  D4  00593              CLRL    -(SP)                              :
                                          50  DD  00595              PUSHL   R0                                 :
            00000000G     EF              03  FB  00597              CALLS   #3, XPO$PUT                        :
                          50              01  D0  0059E  52$:        MOVL    #1, R0                             : 1020
                                          04  005A1              RET                                        :
                          50              02  D0  005A2  53$:        MOVL    #2, R0                             :
                                          04  005A5              RET                                        :
                                          50  D4  005A6  54$:        CLRL    R0                                 : 1034
                                          04  005A8              RET                                        :
```

; Routine Size: 1449 bytes,   Routine Base: $CODE$ + 0000

; 919   1035 1

L 16

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 30
V04-000   get_out_default -- compute output filename from 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (16)

```
 921      1036   1   %sbttl 'get_out_default -- compute output filename from input filename'
 922      1037   1   ROUTINE get_out_default (file_descriptor) : NOVALUE =
 923      1038   1
 924      1039   1   !++
 925      1040   1   ! FUNCTIONAL DESCRIPTION:
 926      1041   1   !
 927      1042   1   !       This routine uses the input file type to compute the
 928      1043   1   !       output file type (i.e., sometimes called extension).
 929      1044   1   !
 930      1045   1   ! FORMAL PARAMETERS:
 931      1046   1   !
 932      1047   1   !       FILE_DESCRIPTOR is a string descriptor for the input file name.
 933      1048   1   !
 934      1049   1   ! IMPLICIT INPUTS:        None
 935      1050   1   !
 936      1051   1   ! IMPLICIT OUTPUTS:       None
 937      1052   1   !
 938      1053   1   ! ROUTINE VALUE:
 939      1054   1   ! COMPLETION CODES:       None
 940      1055   1   !
 941      1056   1   ! SIDE FFFECTS:           None
 942      1057   1   !
 943      1058   1   !--
 944      1059   1
 945      1060   2       BEGIN
 946      1061   2
 947      1062   2       BIND
 948      1063   2           file_spec_stuff = .FILE_DESCRIPTOR : $STR_DESCRIPTOR ();
 949      1064   2
 950      1065   2       LOCAL
 951      1066   2           file_specx,
 952      1067   2           file_spec_lng,
 953      1068   2           name_length,
 954      1069   2           parse_spec_blk : $XPO_SPEC_BLOCK,
 955      1070   2           ptr_u_ext,
 956      1071   2           ptr_i,
 957      1072   2           ptr_o,
 958      1073   2           type_length,
 959      1074   2           u_ext : VECTOR [CH$ALLOCATION (50)];
 960      1075   2
 961      1076   2       file_specx = .file_spec_stuff [STR$A_POINTER];
 962      1077   2       file_spec_lng = .file_spec_stuff [STR$H_LENGTH];
 963      1078   2
 964      1079   2   !The code conditionalized by the %BLISS32 lexical function is
 965      1080   2   !to get around a bug introduced in version 2.0 of VAX/VMS.
 966      1081   2   !Logical names now start with an underscore, and this mucks up
 967      1082   2   !XPORT's file spec parsing.
 968      1083   2   %IF %BLISS(BLISS32) %THEN
 969      1084   2           !Inform the builder of RUNOFF that this kludge is here.
 970    L 1085   2           %MESSAGE ('BLISS32 Patch code included to strip "_" from output filespec');
 971      1086   2           !Ignore the first character of the file spec if it's
 972      1087   2           !an underscore so XPORT; this lets XPORT work correctly.
 973      1088   2           IF  CH$RCHAR(.file_specx) EQL %C'_'
 974      1089   2           THEN
 975      1090   2               !It's an underscore. Do the kludge.
 976      1091   3               BEGIN
 977      1092   3               file_specx = CH$PLUS (.file_specx, 1);
```

16

```
CLH                      file processing interface and command line hand 15-Sep-1984 23:56:16    VAX-11 Bliss-32 V4.0-742          Page 31
V04-000                  get_out_default -- compute output filename from 14-Sep-1984 13:05:41     DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1      (16)
```

```
  978     1093  3                        file_spec_lng = .file_spec_lng - 1;
  979     1094  2                        END;
  980     1095  2    %FI
  981     1096  2
  982     1097  2                    !End of crock code.
  983     1098  2
  984     1099  2                    !Parse the input file spec to get the specified type.
         P  1099  2            $XPO_PARSE_SPEC ( SPEC_BLOCK = parse_spec_blk
  985     1100  2                                    ,FILE_SPEC = ( .file_spec_lng
  986     1101  2                                                 ,.file_specx )   );
  987     1102  2            !Start building the output file spec.
  988     1103  2            !
  989     1104  2            !First set CH$PTRs to the parsed input file name.
  990     1105  2            BEGIN
  991     1106  3            BIND
  992     1107  3                temp = parse_spec_blk [XPO$T_FILE_NAME] : $STR_DESCRIPTOR ();
  993     1108  3
  994     1109  3            name_length = .temp [STR$H_LENGTH];
  995     1110  3            ptr_i = .temp [STR$A_POINTER];
  996     1111  2            END;
  997     1112  2
  998     1113  2            !Set a CH$PTR to where the computed output file spec goes.
  999     1114  2            ptr_o = CH$PTR (def_out_spc);
 1000     1115  2            !Copy the input file name to the output file spec.
 1001     1116  2
 1002     1117  2            INCR i FROM 1 TO .name_length DO
 1003     1118  2                CH$WCHAR_A (CH$RCHAR_A (ptr_i), ptr_o);
 1004     1119  2
 1005     1120  2            def_out_lng = .name_length;                     !Set current length.
 1006     1121  2            BEGIN
 1007     1122  3            BIND
 1008     1123  3                temp = parse_spec_blk [XPO$T_FILE_TYPE] : $STR_DESCRIPTOR ();
 1009     1124  3            type_length = .temp [STR$H_LENGTH];
 1010     1125  3            ptr_i = .temp [STR$A_POINTER];
 1011     1126  2            END;
 1012     1127  2
 1013     1128  2            !Translate the file type to upper case.
 1014     1129  2            !Leave the result in U_EXT.
 1015     1130  2            ptr_u_ext = CH$PTR (u_ext);             !CH$PTR to where file type goes when in upper case.
 1016     1131  2            INCR i FROM 1 TO .type_length DO
 1017     1132  3                BEGIN
 1018     1133  3
 1019     1134  3                LOCAL
 1020     1135  3                    temp;
 1021     1136  3
 1022     1137  3                temp = CH$RCHAR_A (ptr_i);
 1023     1138  3
 1024     1139  4                IF  lower_letter (.temp)
 1025     1140  3                THEN
 1026     1141  3                    !Convert lower case letter to upper case.
 1027     1142  3                    temp = upper_case (.temp);
 1028     1143  3
 1029     1144  3                !Put processed character into file type area.
 1030     1145  3                CH$WCHAR_A (.temp, ptr_u_ext);
 1031     1146  2                END;
 1032     1147  2
 1033     1148  2            ! Search through the various known input filetypes looking for a type that
 1034     1149  2            ! matches what was given.
```

CLH
V04-000

B 1
file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 32
get_out_default -- compute output filename from 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (16)

```
: 1035        1150  2
: 1036        1151  2          ptr_u_ext = .opftop [0];     !Assume it won't be found, and point to ".MEM".
: 1037        1152  2
: 1038        1153  2          IF .ipftyp EQL -1 THEN         ! We haven't yet mapped against IPFTOP
: 1039        1154  3              BEGIN
: 1040        1155  3              INCR i FROM 0 TO (ipftct-1) DO
: 1041        1156  3
: 1042        1157  3                  IF CH$EQL ( .type_length
: 1043        1158  3                             .CH$PTR (u_ext)
: 1044        1159  3                             ,4
: 1045        1160  3                             ,.ipftop [.i]  )
: 1046        1161  3                  THEN
: 1047        1162  3                  !Found a match.  Set a CH$PTR to the matching output file type.
: 1048        1163  4                      BEGIN
: 1049        1164  4                      ptr_u_ext = .opftop [.i];
: 1050        1165  4                      EXITLOOP
: 1051        1166  4                      END
: 1052        1167  3              END
: 1053        1168  2          ELSE
: 1054        1169  2              ptr_u_ext = .opftop[.ipftyp];
: 1055        1170  2
: 1056    U 1171  2  %IF DSRPLUS %THEN
: 1057    U 1172  2      IF  .gca_op_dev EQL op_dev_vt100
: 1058    U 1173  2      THEN
: 1059    U 1174  2          !User said /DEC=VT100, so make .VT1 the default output type.
: 1060    U 1175  2          ptr_u_ext = CH$PTR (UPLIT ('.VT1'));
: 1061      1176  2  %FI
: 1062      1177
: 1063      1178  2  %IF LN01 %THEN
: 1064      1179  3      IF (.gca_op_dev EQL op_dev_ln01
: 1065      1180  3         OR .gca_op_dev EQL op_dev_ln01e)
: 1066      1181  2      THEN
: 1067      1182  2          !User said /DEVICE=LN01[e], so make .LNI the default output type.
: 1068      1183  2          ptr_u_ext = CH$PTR (UPLIT ('.LNI'));
: 1069      1184  2  %FI
: 1070      1185  2
: 1071    U 1186  2  %IF FLIP %THEN
: 1072    U 1187  2      IF  (.gca_op_dev EQL op_dev_flip)
: 1073    U 1188  2      THEN
: 1074    U 1189  2          !User said /DEC=FLIP; this overrides a /DEC=VT100 (if also given).
: 1075    U 1190  2          ptr_u_ext = CH$PTR (UPLIT ('.BFL'));
: 1076      1191  2  %FI
: 1077      1192  2
: 1078      1193  2      !Copy the file type to the output file spec area.
: 1079      1194  2      !Note that if there was no match, then PTR_U_EXT points to ".MEM"
: 1080      1195  2
: 1081      1196  2      INCR i FROM 1 TO 4 DO
: 1082      1197  2          CH$WCHAR_A (CH$RCHAR_A (ptr_u_ext), ptr_o);
: 1083      1198  2
: 1084      1199  2      def_out_lng = .def_out_lng + 4;          !Update file spec length.
: 1085      1200  1      END;                                     !End of get_out_default


                                                                 .PSECT   $PLIT$,NOWRT,NOEXE,2

                                        49 4E 4C 2E  0003C P.AAK:  .ASCII  \.LNI\
```

```
                                                          .EXTRN  XPO$PARSE_SPEC

                                                          .PSECT  $CODE$,NOWRT,2

                                    07FC 00000 GET_OUT_DEFAULT:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10              ; 1037
                   5A 00000000G EF 9E 00002               MOVAB   GCA+208, R10
                   59 00000000G EF 9E 00009               MOVAB   OPFTOP, R9
                   58 00000000' EF 9E 00010               MOVAB   DEF_OUT_LNG, R8
                   5E          FF7C CE 9E 00017           MOVAB   -132(SP), SP
                   50          AC D0 0001C                MOVL    FILE_DESCRIPTOR, R0           ; 1063
                   51    04    A0 D0 00020                MOVL    4(R0), FILE_SPECX            ; 1076
                   50          60 3C 00024                MOVZWL  (R0), FILE_SPEC_LNG          ; 1077
            5F 8F  61          91 00027                   CMPB    (FILE_SPECX), #95           ; 1088
                   04          12 0002B                   BNEQ    1$
                   51          D6 0002D                   INCL    FILE_SPECX                   ; 1092
                   50          D7 0002F                   DECL    FILE_SPEC_LNG                ; 1093
            6E     50          B0 00031 1$:               MOVW    FILE_SPEC_LNG, $STR$FILE_SPEC ; 1101
            02 AE  0E          90 00034                   MOVB    #14, $STR$FILE_SPEC+2
            03 AE  01          90 00038                   MOVB    #1, $STR$FILE_SPEC+3
            04 AE  51          D0 0003C                   MOVL    FILE_SPECX, $STR$FILE_SPEC+4
               00000000G EF    9F 00040                   PUSHAB  XPO$FAILURE
                   7E          D4 00046                   CLRL    -(SP)
            7E     01          CE 00048                   MNEGL   #1, -(SP)
               48 AE           9F 0004B                   PUSHAB  PARSE_SPEC_BLK
               10 AE           9F 0004E                   PUSHAB  $STR$FILE_SPEC
     00000000G EF  05          FB 00051                   CALLS   #5, XPO$PARSE_SPEC
            51  5C AE          3C 00058                   MOVZWL  TEMP, NAME_LENGTH            ; 1109
            52  60 AE          D0 0005C                   MOVL    TEMP+4, PTR_I               ; 1110
            56  04 A8          9E 00060                   MOVAB   DEF_OUT_SPC, PTR_O          ; 1114
                   50          D4 00064                   CLRL    I                            ; 1118
                   03          11 00066                   BRB     3$
            86     50          82 90 00068 2$:            MOVB    (PTR_I)+, (PTR_O)+
     F9     50     51          F3 0006B 3$:               AOBLEQ  NAME_LENGTH, I, 2$
            68     51          D0 0006F                   MOVL    NAME_LENGTH, DEF_OUT_LNG     ; 1120
            57  64 AE          3C 00072                   MOVZWL  TEMP, TYPE_LENGTH           ; 1124
            52  68 AE          D0 00076                   MOVL    TEMP+4, PTR_I               ; 1125
            55  08 AE          9E 0007A                   MOVAB   U_EXT, PTR_O_EXT            ; 1130
                   51          D4 0007E                   CLRL    I                            ; 1131
                   1B          11 00080                   BRB     6$
            50     82          9A 00082 4$:               MOVZBL  (PTR_I)+, TEMP              ; 1137
     00000061 8F  50          D1 00085                   CMPL    TEMP, #97                   ; 1139
                   0C          19 0008C                   BLSS    5$
     0000007A 8F  50          D1 0008E                   CMPL    TEMP, #122
                   03          14 00095                   BGTR    5$
            50     20          C2 00097                   SUBL2   #32, TEMP                   ; 1142
            50     85          90 0009A 5$:               MOVB    TEMP, (PTR_U_EXT)+          ; 1145
     E1     51     57          F3 0009D 6$:               AOBLEQ  TYPE_LENGTH, I, 4$          ; 1131
            55     69          D0 000A1                   MOVL    OPFTOP, PTR_U_EXT           ; 1151
            50 00000000G EF    D0 000A4                   MOVL    IPFTYP, R0                  ; 1153
     FFFFFFFF 8F  50          D1 000AB                   CMPL    R0, #-1
                   26          12 000B2                   BNEQ    9$
                   54          01 CE 000B4                 MNEGL   #1, I                      ; 1155
                   17          11 000B7                   BRB     8$
            50 00000000GEF44   D0 000B9 7$:               MOVL    IPFTOP[I], R0              ; 1160
     04    00    08 AE         57 2D 000C1               CMPC5   TYPE_LENGTH, U_EXT, #0, #4, (R0)  ; 1157
```

```
                                        D 1
CLH              file processing interface and command line hand 15-Sep-1984 23:56:16    VAX-11 Bliss-32 V4.0-742        Page 34
V04-000          get_out_default -- compute output filename from 14-Sep-1984 13:05:41    DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (16)
```

```
                          60      000C7
                  06   12 000C8          BNEQ    8$
            55         6944 DO 000CA      MOVL    OPFTOP[I], PTR_U_EXT
                  0E   11 000CE          BRB     10$
      E1    54 00000000G 8F F3 000D0 8$: AOBLEQ  #IPFTCT-1, I, 7$
                  04   11 000D8          BRB     10$
            55         6940 DO 000DA 9$: MOVL    OPFTOP[R0], PTR_U_EXT
 04         6A    04   04 ED 000DE 10$:  CMPZV   #4, #4, GCA+208, #4
                  07   13 000E3          BEQL    11$
 05         6A    04   04 ED 000E5       CMPZV   #4, #4, GCA+208, #5
                  07   12 000EA          BNEQ    12$
            55 00000000' EF 9E 000EC 11$: MOVAB  P.AAK, PTR_U_EXT
            50         01 DO 000F3 12$:   MOVL    #1, I
            86         85 90 000F6 13$:   MOVB    (PTR_U_EXT)+, (PTR_O)+
      F9    50         04 F3 000F9        AOBLEQ  #4, I, -13$
            68         04 CO 000FD        ADDL2   #4, DEF_OUT_LNG
                       04 00100           RET
```

```
; Routine Size: 257 bytes,   Routine Base: $CODE$ + 05A9


; 1086        1201  1
```

```
                                                                  1164
                                                                  1163
                                                                  1157
                                                                  1154
                                                                  1169
                                                                  1179
                                                                  1180
                                                                  1183
                                                                  1197
                                                                  1199
                                                                  1200
```

E 1

CLH   file processing interface and command line hand 15-Sep-1984 23:56:16   VAX-11 Bliss-32 V4.0-742   Page 35
V04-000   FBWAIT -- performs user syncronization for /PAU 14-Sep-1984 13:05:41   DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (17)

```
: 1088        1202  1  %sbttl 'FBWAIT -- performs user syncronization for /PAUSE O/P w/FF'
: 1089        1203  1  GLOBAL ROUTINE fbwait : NOVALUE =
: 1090        1204  1
: 1091        1205  1  !++
: 1092        1206  1  ! FUNCTIONAL DESCRIPTION:
: 1093        1207  1  !
: 1094        1208  1  !       Issues some BELLs (^Gs) and a FORMFEED (^L) and waits for
: 1095        1209  1  !       the RUNOFF user to input a single character.
: 1096        1210  1  !
: 1097        1211  1  ! FORMAL PARAMETERS:        None
: 1098        1212  1  !
: 1099        1213  1  ! IMPLICIT INPUTS:          None
: 1100        1214  1  !
: 1101        1215  1  ! IMPLICIT OUTPUTS:         None
: 1102        1216  1  !
: 1103        1217  1  ! ROUTINE VALUE:
: 1104        1218  1  ! COMPLETION CODES:         None
: 1105        1219  1  !
: 1106        1220  1  ! SIDE EFFECTS:             None
: 1107        1221  1  !
: 1108        1222  1  !--
: 1109        1223  1
: 1110        1224  2      BEGIN
: 1111      P 1225  2      $XPO_GET (IOB = tsiiob
: 1112      P 1226  2                ,PROMPT = (3, CH$PTR (UPLIT (
: 1113      P 1227  2                          %STRING (BELL, BELL, %CHAR (%O'14')))) )
: 1114      P 1228  2                ,CHARACTERS = 1
: 1115        1229  2                );
: 1116        1230  2
: 1117        1231  2      !Send a carriage return, so text starts at the left margin
: 1118      P 1232  2      $XPO_PUT (IOB = tsiiob
: 1119      P 1233  2                , STRING = (1, CH$PTR(UPLIT(%STRING(%CHAR(%O'15')))))
: 1120        1234  2                );
: 1121        1235  2
: 1122        1236  1      END;                                      !End of FBWAIT


                                             .PSECT  $PLIT$,NOWRT,NOEXE,2

                   00  0C  07  07  00040 P.AAP:  .ASCII  <7><7><12><0>
                   00  00  00  0D  00044 P.AAT:  .ASCII  <13><0><0><0>

                                             .EXTRN  XST$FREE_TEMP

                                             .PSECT  $CODE$,NOWRT,2

                                  001C 00000     .ENTRY  FBWAIT, Save R2,R3,R4          : 1203
                   54 00000000G  EF  9E 00002     MOVAB   XPO$FAILURE, R4
                   53 00000000G  EF  9E 00009     MOVAB   IOB$+36, R3
                   5E            08  C2 00010     SUBL2   #8, SP
                   6E            03  B0 00013     MOVW    #3, $STR$STRING              : 1229
                02 AE            0E  90 00016     MOVB    #14, $STR$STRING+2
                03 AE            01  90 0001A     MOVB    #1, $STR$STRING+3
                04 AE 00000000'  EF  9E 0001E     MOVAB   P.AAP, $STR$STRING+4
                   7E            D4 00026     CLRL    -(SP)
                04 AE            9F 00028     PUSHAB  $STR$STRING
```

```
                                        F-1
CLH                 file processing interface and command line hand 15-Sep-1984 23:56:16     VAX-11 Bliss-32 V4.0-742    Page 36
V04-000             FBWAIT -- performs user syncronization for /PAU 14-Sep-1984 13:05:41     DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1    (17)
```

```
                              7E  D4 0002B              CLRL     -(SP)
            00000000G  EF     03  FB 0002D              CALLS    #3, XST$FORMAT
                       52     50  D0 00034              MOVL     R0, R2
                       50     63  D0 00037              MOVL     IOB$+36, R0
                              09  13 0003A              BEQL     1$
                              50  DD 0003C              PUSHL    R0
            00000000G  EF     01  FB 0003E              CALLS    #1, XST$FREE_TEMP
                       63     52  D0 00045    1$:       MOVL     R2, IOB$+36
                    10 A3     01  B0 00048              MOVW     #1, IOB$+52
                    12 A3     0E  90 0004C              MOVB     #14, IOB$+54
                    08 A3     06  90 00050              MOVB     #6, IOB$+44
                              54  DD 00054              PUSHL    R4
                              7E  D4 00056              CLRL     -(SP)
                    DC A3     9F  00058              PUSHAB   IOB$
            00000000G  EF     03  FB 0005B              CALLS    #3, XPO$GET
                       6E     01  B0 00062              MOVW     #1, $IOB$OUTPUT
                    02 AE     0E  90 00065              MOVB     #14, $IOB$OUTPUT+2
                    03 AE     01  90 00069              MOVB     #1, $IOB$OUTPUT+3
                    04 AE 00000000' EF 9E 0006D         MOVAB    P.AAT, $IOB$OUTPUT+4
                    20 A3     6E  9E 00075              MOVAB    $IOB$OUTPUT, IOB$+68
                    08 A3     07  90 00079              MOVB     #7, IOB$+44
                              54  DD 0007D              PUSHL    R4
                              7E  D4 0007F              CLRL     -(SP)
                    DC A3     9F  00081              PUSHAB   IOB$
            00000000G  EF     03  FB 00084              CALLS    #3, XPO$PUT
                              04  0008B              RET
```

```
; Routine Size: 140 bytes,    Routine Base:  $CODE$ + 06AA
```

```
; 1123        1237  1
```

```
CLH                    file processing interface and command line hand 15-Sep-1984 23:56:16    VAX-11 Bliss-32 V4.0-742      Page  37
V04-000                BWAIT -- performs user syncronization for /PAUS 14-Sep-1984 13:05:41    DISK$VMSMASTER:[RUNOFF.SRC]CLH.BLI;1   (18)
```

```
:  1125      1238   1   %sbttl 'BWAIT -- performs user syncronization for /PAUSE O/P w/o FF'
:  1126      1239   1   GLOBAL ROUTINE bwait : NOVALUE =
:  1127      1240   1
:  1128      1241   1   !++
:  1129      1242   1   !  FUNCTIONAL DESCRIPTION:
:  1130      1243   1   !
:  1131      1244   1   !        This routine is just like FBWAIT, except that no FORMFEED is issued.
:  1132      1245   1   !
:  1133      1246   1   !  FORMAL PARAMETERS:       None
:  1134      1247   1   !
:  1135      1248   1   !  IMPLICIT INPUTS:         None
:  1136      1249   1   !
:  1137      1250   1   !  IMPLICIT OUTPUTS:        None
:  1138      1251   1   !
:  1139      1252   1   !  ROUTINE VALUE:
:  1140      1253   1   !  COMPLETION CODES:        None
:  1141      1254   1   !
:  1142      1255   1   !  SIDE EFFECTS:            None
:  1143      1256   1   !
:  1144      1257   1   !--
:  1145      1258   1
:  1146      1259   2       BEGIN
:  1147      1260   2
:  1148      1261   2       EXTERNAL
:  1149      1262   2           tsiiob: $XPO_IOB();
:  1150      1263   2
:  1151    P 1264   2       $XPO_GET (IOB = tsiiob
:  1152    P 1265   2               ,PROMPT = (4, CH$PTR(UPLIT(%STRING(BELL,DEL,BELL,DEL))) )
:  1153    P 1266   2               ,CHARACTERS = 1
:  1154      1267   2               );
:  1155      1268   2
:  1156      1269   2       !Send a carriage return, so text starts at the left margin
:  1157    P 1270   2       $XPO_PUT (IOB = tsiiob
:  1158    P 1271   2               ,STRING = (1, CH$PTR(UPLIT(%STRING(%CHAR(%O'15')))))
:  1159      1272   2               );
:  1160      1273   1       END;                                    !End of BWAIT
```

```
                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                              7F  07  7F  07  00048 P.AAY:  .ASCII  <7><127><7><127>
                              00  00  00  0D  0004C P.ABC:  .ASCII  <13><0><0><0>


                                                          .PSECT  $CODE$,NOWRT,2

                                          001C 00000       .ENTRY  BWAIT, Save R2,R3,R4                    : 1239
                    54 00000000G  EF  9E 00002            MOVAB   XPO$FAILURE, R4
                    53 00000000G  EF  9E 00009            MOVAB   IOB$+36, R3
                    5E            08  C2 00010            SUBL2   #8, SP
                    6E            04  B0 00013            MOVW    #4, $STR$STRING                          : 1267
                 02 AE           0E  90 00016            MOVB    #14, $STR$STRING+2
                 03 AE           01  90 0C01A            MOVB    #1, $STR$STRING+3
                 04 AE 00000000' EF  9E 0001E            MOVAB   P.AAY, $STR$STRING+4
                    7E            D4 00026            CLRL    -(SP)
```

```
                                04    AE  9F 00028              PUSHAB   $STR$STRING
                                      7E  D4 0002B              CLRL     -(SP)
                  00000000G  EF  03    FB 0002D              CALLS    #3, XST$FORMAT
                            52        50  D0 00034              MOVL     R0, R2
                            50        63  D0 00037              MOVL     IOB$+36, R0
                                      09  13 0003A              BEQL     1$
                                      50  DD 0003C              PUSHL    R0
                  00000000G  EF  01    FB 0003E              CALLS    #1, XST$FREE_TEMP
                            63        52  D0 00045  1$:         MOVL     R2, IOB$+36
                      10    A3        01  B0 00048              MOVW     #1, IOB$+52
                      12    A3        0E  90 0004C              MOVB     #14, IOB$+54
                      08    A3        06  90 00050              MOVB     #6, IOB$+44
                                      54  DD 00054              PUSHL    R4
                                      7E  D4 00056              CLRL     -(SP)
                            DC    A3  9F 00058              PUSHAB   IOB$
                  00000000G  EF  03    FB 0005B              CALLS    #3, XPO$GET
                            6E        01  B0 00062              MOVW     #1, $IOB$OUTPUT
                      02    AE        0E  90 00065              MOVB     #14, $IOB$OUTPUT+2
                      03    AE        01  90 00069              MOVB     #1, $IOB$OUTPUT+3
                04    AE  00000000'  EF  9E 0006D              MOVAB    P.ABC, $IOB$OUTPUT+4
                      20    A3        6E  9E 00075              MOVAB    $IOB$OUTPUT, IOB$+68
                      08    A3        07  90 J0079              MOVB     #7, IOB$+44
                                      54  DD 0007D              PUSHL    R4
                                      7E  D4 0007F              CLRL     -(SP)
                            DC    A3  9F 00081              PUSHAB   IOB$
                  00000000G  EF  03    FB 00084              CALLS    #3, XPO$PUT
                                      04  0008B              RET
```

; Routine Size: 140 bytes,   Routine Base: $CODE$ + 0736

```
; 1161              1274  1
; 1162              1275  1 END                                                          !End of module
; 1163              1276  0 ELUDOM
```

         PSECT SUMMARY

Name	Bytes	Attributes
OWN	68	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$PLIT$	80	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$CODE$	1986	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

       Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]XPORT.L32;1	590	168	28	252	00:00.1

; _255DUA28:[RUNOFF.SRC]DSRLIB.L32;1   1248   54   4   86   00:00.2

;   COMMAND QUALIFIERS

;   BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CLH/OBJ=OBJ$:CLH MSRC$:CLH/UPDATE=(ENH$:CLH)
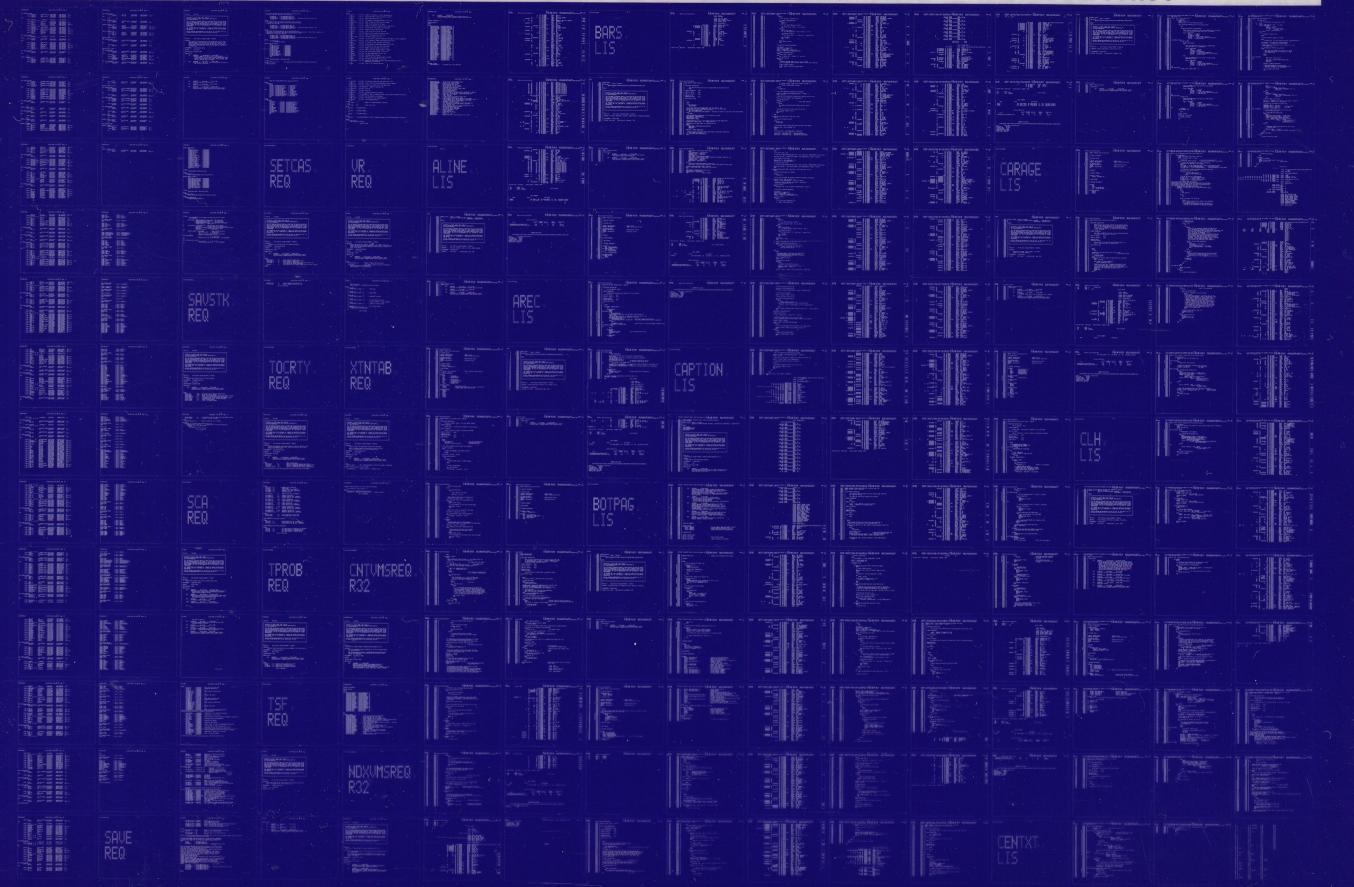
; Size: 1986 code + 148 data bytes
; Run Time: 01:03.8
; Elapsed Time: 01:54.9
; Lines/CPU Min: 1200
; Lexemes/CPU-Min: 73280
; Memory Used: 465 pages
; Compilation Complete

BARS
LIS

SETCAS
REQ

VR
REQ

ALINE
LIS

CARAGE
LIS

SAVSTK
REQ

AREC
LIS

TOCRTY
REQ

XTNTAB
REQ

CAPTION
LIS

CLH
LIS

SCA
REQ

BOTPAG
LIS

TPROB
REQ

CNTVMSREQ
R32

TSF
REQ

NDXVMSREQ
R32

SAVE
REQ

CENTXT
LIS

CONVBB
LIS

CNVDAT
LIS

CONVBR
LIS

CNTVMSMSG
LIS

CONTENTS
LIS

CNTVMS
LIS

CNTVRS
LIS

CONTRL
LIS

CONVBL
LIS